Bergische Universität Wuppertal
Fakultät für Mathematik und Naturwissenschaften

**Dissertation**

# Efficient Computation of the Action of Matrix Rational Functions and Laplace Transforms

eingereicht von
Manuel Tsolakis, M. Sc.
zur Erlangung des Grades eines Doktors der Naturwissenschaften

Wuppertal, 28. April 2023

Betreut durch Prof. Dr. Andreas Frommer,
Dr. Karsten Kahl und
Dr. Marcel Schweitzer.

七転び八起き

Fall down seven times, get up eight.

*(Japanese Proverb)*

**Acknowledgments**

This endeavor would not have been possible without Prof. Dr. Andreas Frommer as he continuously helped me grow as a mathematician by asking the right questions, giving the right answers or pointing in the right direction. Moreover, he provided the initial research ideas and offered me the doctorate position in the first place.

I am also grateful to Dr. Karsten Kahl for directly guiding me, answering my questions, offering me new (mathematical) viewpoints or just having inspiring discussions.

Special thanks also go to Dr. Marcel Schweitzer whose mathematical expertise (especially in Krylov subspaces) and willingness to dig through MATLAB code helped a lot. I also thank him for the soft drink in Manchester the locals called “beer”.

I thank all three of them for their unique personalities and for dedicating much of their time—even when it was scarce—to me and my work.

# Contents

# Notation

Throughout this thesis, vectors are denoted by lower-case letters in bold font whereas matrices are denoted by upper-case letters. While most of our notation can be considered standard, we mention the following to avoid potential confusion:

Notation regarding scalars

| | |
|---|---|
| $\overline{s}$ | complex conjugate of $s$ |
| $\lfloor s \rfloor$ | floor function of $s$; largest integer $n \leq s$ |
| $\mathbb{C}_\infty$ | set of extended complex numbers; $\mathbb{C} \cup \{\infty\}$ |
| $\mathbb{R}^+$ | strictly positive real axis; $(0, \infty)$ |
| $\mathbb{R}_0^+$ | non-negative real axis; $\mathbb{R}^+ \cup \{0\}$ |
| $\mathrm{Im}(s)$ | imaginary part of $s$ |
| $\mathrm{Re}(s)$ | real part of $s$ |
| $\mathrm{K}_{i=1}^{\alpha} \left(\frac{c_i}{b_i}\right)$ | continued fraction (see Definition 2.38) |

Notation regarding functions

| | |
|---|---|
| $f^{[k]}$ | $k$th derivative of $f$ |
| $L^1(E)$ | set of functions that are integrable in $E$ |
| $L^1_{\mathrm{loc}}(E)$ | set of functions that are locally integrable in $E$ |
| $\mathcal{L}\{\hat{f}\}$ | Laplace transform of $\hat{f}$ (see Definition 2.1) |
| $\mathcal{O}(f)$ | big O notation |
| $\alpha\{\hat{f}\}$ | abscissa of existence of $\mathcal{L}\{\hat{f}\}$ (see Definition 2.3) |
| $\Delta_{[\mu_1,\ldots,\mu_j]}\{f\}$ | divided difference of $f$ with nodes $\mu_1, \ldots, \mu_j$ |
| $\chi_E$ | characteristic function of the set $E$ |
| $\omega\{\hat{f}\}$ | exponential order of $\hat{f}$ (see Definition 2.5) |

Notation regarding matrices

| | |
|---|---|
| $\boldsymbol{e}_i$ | $i$th column of the identity matrix |
| $I_n$ | identity matrix of size $n$ (without index if obvious) |
| $\mathfrak{J}(A)^{-1}$ | approximation to $A^{-1}$ (see Section 2.5.2) |
| $\mathcal{K}_m(A, \boldsymbol{b})$ | $m$th Krylov subspace of $A$ and $\boldsymbol{b}$ (see Definition 2.59) |
| $\kappa(A)$ | condition number of $A$ (w.r.t. the 2-norm) |
| $\operatorname{spec}(A)$ | set of eigenvalues of the matrix $A$ |
| $\operatorname{specp}(T)$ | set of eigenvalues of the pencil $T(s)$ (see Definition 2.82) |
| $T_m(A)$ | CF-matrix (see Definition 3.1) |
| $\mathcal{W}(A)$ | numerical range of $A$ |
| $A^{\mathsf{H}}$ | conjugate transpose of $A$ |
| $A^{\mathsf{T}}$ | transpose of $A$ |
| $A \otimes B$ | Kronecker product |
| $A \oplus B$ | Kronecker sum; $A \otimes I + I \otimes B$ |
| $A \odot B$ | element-wise (Hadamard) product; $[a_{ij}b_{ij}]$ |
| $A \hat{\oplus} B$ | direct sum; $\begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix}$ |

# 1. Introduction

Given a matrix $A \in \mathbb{C}^{n\times n}$, a vector $\boldsymbol{b} \in \mathbb{C}^n$ and a function $f : \mathbb{C} \to \mathbb{C}$, the *action of a matrix function* is defined as

$$f(A)\boldsymbol{b},$$

i.e., as the product of the matrix function $f(A) \in \mathbb{C}^{n\times n}$ and the vector $\boldsymbol{b}$. An important example is the exponential function $f(s) = \exp(cs)$ because it arises, e.g., from solving (matrix) differential equations [73]. Other examples include the square root in machine learning [76] and other fields [7, 59], the sign function with applications in quantum chromodynamics (theoretical physics) [32] and the logarithm, which appears as the "entropy" for example in quantum statistical mechanics [10]. A naive approach for computing $f(A)\boldsymbol{b}$ is to first evaluate $f(A)$ and then multiply this matrix with $\boldsymbol{b}$. Of course, for increasing $n$, this becomes prohibitively expensive in terms of required arithmetic operations or memory.

Although it is usually not considered a matrix function, the most common case is the inverse $f(s) = s^{-1}$, for which the action is the solution $\boldsymbol{x}$ of the linear system

$$A\boldsymbol{x} = \boldsymbol{b}.$$

This special case has many properties that are not present for other choices of $f$ (e.g., the multiplicative inverse of $f(s)$ coincides with its inverse function, $f(s)^{-1} = f^{-1}(s)$) but can be exploited for very efficient algorithms to compute $A^{-1}\boldsymbol{b}$ that can be used even for very large $n$. Important examples are the conjugate gradient method (CG) and the generalized minimal residual method (GMRES). Both belong to the class of Krylov subspace methods that project the linear system onto a subspace that is increased in each iteration. Another important class is the class of multigrid methods, which combine cheap iterative methods with a subspace correction. In contrast to Krylov subspace methods, the subspace is constructed beforehand based on the structure of $A$ and remains fixed.

Extending these methods to other functions $f$ is not always trivial. One approach is to first replace $f$ by a rational approximation $r(s) \approx f(s)$. One can then use the partial fraction expansion, i.e.,

$$r(s) = \sum_{i=1}^{m} \frac{w_i}{s + \tau_i}$$

assuming that $r$ has only simple poles, to express $r(A)\boldsymbol{b}$ by the linear systems

$$(A + \tau_i I)\boldsymbol{x} = \boldsymbol{b}. \tag{1.1}$$

These systems can be solved using the standard methods mentioned above. Indeed, specialized Krylov subspace methods allow one to solve all of these systems approximately for the price of one. On the other hand, multigrid methods need to solve each system separately and are thus at a disadvantage.

Even rarer is the application of multigrid methods for arbitrary $f$ for there is no general multigrid method for $f(A)\boldsymbol{b}$ available. In contrast, the simplest Krylov subspace method, the Arnoldi method, can be used for more or less any $f$. One problem that Krylov subspace methods face, however, is that the computational cost increases with each iteration, i.e., the size of the current subspace. Thus, the subspace needs to be kept small in some cases—for example, because of an insufficient amount of available memory. One approach is to restart the method, which is already widely known for GMRES. Efficient and stable restarting algorithms require a suitable error representation, which is available only for certain classes of functions, however. In [40], such an error representation and a restarting algorithm based on it were presented for analytic functions. While this covers most of the relevant functions, the algorithm needs additional input from the user. In particular, one needs to choose a contour that depends on $f$ and $A$, which is usually not a trivial task. If $f$ is a Stieltjes function and $A$ positive definite (in the sense that $\boldsymbol{x}^{\mathsf{H}}A\boldsymbol{x} > 0$ for every $\boldsymbol{x} \neq 0$), however, no hand-chosen contour or similar input is necessary. In that sense, the algorithm can be considered a "black-box" method (only) for Stieltjes functions.

In this thesis, we develop new ways for computing $f(A)\boldsymbol{b}$. In addition to the theoretical groundwork in Chapter 2, this thesis treats two topics: In Chapter 3, we present a new representation of $r(A)\boldsymbol{b}$, where $r$ is a rational function given as a continued fraction. We construct a block tridiagonal matrix $T$, which we call the *CF-matrix*. We then show that we only need to solve one linear system with $T$ to obtain $r(A)\boldsymbol{b}$. However, each block of $T$ has the same size as $A$, i.e., $T \in \mathbb{C}^{mn\times mn}$ is much larger than $A$. Thus, we need to carefully investigate whether the CF-matrix yields any computational advantage compared to existing methods for $r(A)\boldsymbol{b}$, in particular Krylov subspace and multigrid methods for Eq. (1.1). To this end, we present theoretical results and numerical experiments.

Chapter 4 is the second part of this thesis. Here, we present a new error representation for the restarted Arnoldi method if $f$ is a Laplace transform or a complete Bernstein function. Based on this representation, we discuss the necessary steps to develop a new restarting algorithm that is efficient and stable. While our functions are analytic and thus could be treated by the algorithm from [40], our method does not need a hand-chosen contour. We only need that the numerical range of $A$ is "far enough to the right", which for many $f$ reads as $A$ being positive definite. The class of Stieltjes functions is also covered by our algorithm, so it can be considered a generalization of the "black-box" approach of [40]. Moreover, we present a new error bound for the restarted Arnoldi method, which proves that our algorithm converges at least linearly in some cases. Numerical experiments illustrating the performance of our algorithm and error bounds complete the chapter.

Many results in this thesis (though not all) have already been published: Parts of Chapter 3 are based on [CF], whereas parts of Chapter 4 are based on [L]. However, we also present new results accompanied by fitting numerical experiments. We give more details at the beginning of the chapters.

# 2. Review of basic material

We start as usual with a review of known results. Throughout this thesis, we assume familiarity with basic concepts of linear algebra and Lebesgue integration[1].

As we are interested in computing $f(A)\boldsymbol{b}$, we start by introducing the classes of functions $f$ most relevant in this thesis in Section 2.1. In Section 2.2, we describe continued fractions, which we use to represent rational functions. Of course, we need to know how $f(A)$ is defined in the first place if we want to compute it. Thus, we state its definition in Section 2.3. While there is a lot that could be said about methods that compute $f(A)$, we are interested only in $f(A)\boldsymbol{b}$, for which more efficient algorithms exist. The two main ideas that we consider here are the following:

- Project $f(A)\boldsymbol{b}$ onto a subspace and prolongate the projected solution back to the original space. One particular way of doing this is using Krylov subspaces, which is described in Section 2.4 and is the foundation of Chapter 4.

- Approximate the function $f$ by a function that can be computed more easily, i.e., $f \approx r \implies f(A)\boldsymbol{b} \approx r(A)\boldsymbol{b}$. This is the foundation of Chapter 3, where $r(s) = \frac{p(s)}{q(s)}$ is a rational function that can be represented as a continued fraction. We consider how to compute $r(A)\boldsymbol{b}$ when $r(s)$ is a rational function in Section 2.5.

We end the chapter by giving a short description of matrix pencils in Section 2.6 as they are required in Chapter 3.

## 2.1. Relevant classes of functions

In this section, we discuss the classes of functions that we consider in Chapters 3 and 4, but we make references in later parts of this chapter as well. In addition to their definitions, relevant properties and connections are included.

Laplace transforms (Section 2.1.1) are the cornerstone of our discussion here and in Chapter 4. We establish their connection to other classes of functions. We drew inspiration from [6, 25, 80, 94] for this section.

[1] For more information about Lebesgue integration, see for example [77].

### 2.1.1. Laplace transforms

**Definition 2.1.** The *Laplace transform* $\mathcal{L}_t\{\hat{f}(t)\}(s)$ of a measurable function $\hat{f}$ is defined by the proper Lebesgue integral

$$\mathcal{L}_t\{\hat{f}(t)\}(s) \coloneqq \int_0^\infty \exp(-ts)\hat{f}(t)\,\mathrm{d}t, \quad s \in \mathbb{C}, \tag{2.1}$$

whenever it has a finite value. In case the integration variable is clear, we write $\mathcal{L}\{\hat{f}(t)\}(s)$ or just $\mathcal{L}\{\hat{f}\}(s)$.

Let us first consider necessary conditions for the existence of the Laplace transform, i.e., for a finite integral in Eq. (2.1). One such condition is that $\hat{f}$ is locally integrable:

**Lemma 2.2.** *Let $\hat{f}$ (and $s$) be such that*

$$\int_0^\infty \exp(-ts)\hat{f}(t)\,\mathrm{d}t < \infty.$$

*Then*

$$\int_0^T \hat{f}(t)\,\mathrm{d}t < \infty$$

*for every finite $T \geq 0$, i.e., $\hat{f} \in L^1_{\mathrm{loc}}(\mathbb{R}_0^+)$.*

*Proof.* By the definition of the Lebesgue integral, we have

$$\int_0^\infty \exp(-ts)\hat{f}(t)\,\mathrm{d}t < \infty \iff \int_0^\infty \exp(-t\,\mathrm{Re}(s))|\hat{f}(t)|\,\mathrm{d}t < \infty.$$

The Monotone Convergence Theorem (see, e.g., [77, Theorem 1.26]) tells us that the integral on the right side is equal to

$$\int_0^\infty \exp(-t\,\mathrm{Re}(s))|\hat{f}(t)|\,\mathrm{d}t = \lim_{T\to\infty} \int_0^T \exp(-t\,\mathrm{Re}(s))|\hat{f}(t)|\,\mathrm{d}t.$$

The integrals in the limit are non-decreasing, so the limit can only converge to a finite value if

$$\int_0^T \exp(-t\,\mathrm{Re}(s))|\hat{f}(t)|\,\mathrm{d}t < \infty$$

for every finite $T \geq 0$. It easily follows ([77, 1.24(a)]) that

$$\int_0^T \exp(-t\,\mathrm{Re}(s))|\hat{f}(t)|\,\mathrm{d}t \geq c_s \int_0^T |\hat{f}(t)|\,\mathrm{d}t, \quad c_s = \begin{cases} \exp(-T\,\mathrm{Re}(s)), & \mathrm{Re}(s) > 0, \\ 1, & \mathrm{Re}(s) \leq 0. \end{cases}$$

The integral on the right side is finite for every $T$ if and only if $\hat{f} \in L^1_{\mathrm{loc}}(\mathbb{R}_0^+)$. □

The region of existence of Laplace transforms is generally a right half-plane in the complex plane. We can thus describe the region of existence by a single number.

**Definition 2.3.** We call the number

$$\alpha_t\{\hat{f}(t)\} \equiv \alpha\{\hat{f}\} \coloneqq \inf\{\mathrm{Re}(s) : \mathcal{L}\{\hat{f}\}(s) \text{ exists}\}$$

the *abscissa of existence* of $\mathcal{L}\{\hat{f}\}$.

**Theorem 2.4** ([25, Theorem 3.3])**.** *The exact region of existence of the Laplace transform of $\hat{f}$ is either the open half-plane* $\mathrm{Re}(s) > \alpha\{\hat{f}\}$ *or the closed half-plane* $\mathrm{Re}(s) \geq \alpha\{\hat{f}\}$.

The abscissa of existence can be bounded if $\hat{f}$ grows (at most) exponentially.

**Definition 2.5.** We call

$$\omega_t\{\hat{f}(t)\} \equiv \omega\{\hat{f}\} \coloneqq \inf\{\omega \in \mathbb{R} : \hat{f}(t) = \mathcal{O}(\exp(t\omega)) \text{ as } t \to \infty\}$$

the *exponential order* of $\hat{f}$.

**Lemma 2.6** (cf. [6, Eq. (1.10)])**.** *Let $\hat{f} \in L^1_{\mathrm{loc}}(\mathbb{R}_0^+)$. Then $\alpha\{\hat{f}\} \leq \omega\{\hat{f}\}$.*

*Proof.* If $\omega\{\hat{f}\} = \infty$, then the statement is trivial. If $\omega\{\hat{f}\} < \infty$, then for any $\omega > \omega\{\hat{f}\}$, we can write $|\hat{f}(t)| \leq c\exp(t\omega)$ for $t \geq T$ and some $T \geq 0$, $c \geq 0$. Now,

$$\mathcal{L}\{\hat{f}\}(s) = \int_0^T \exp(-ts)\hat{f}(t)\,\mathrm{d}t + \int_T^\infty \exp(-ts)\hat{f}(t)\,\mathrm{d}t.$$

The first integral has a finite value because $\hat{f} \in L^1_{\mathrm{loc}}(\mathbb{R}_0^+)$. We see from

$$\int_T^\infty |\exp(-ts)\hat{f}(t)|\,\mathrm{d}t \leq c\int_T^\infty \exp(-t\,\mathrm{Re}(s))\exp(t\omega)\,\mathrm{d}t$$

that the second integral has a finite value if $\mathrm{Re}(s) > \omega > \omega\{\hat{f}\}$. Since $\alpha\{\hat{f}\}$ is by definition the smallest number $\alpha$ such that $\mathcal{L}\{\hat{f}\}(s)$ exists for $\mathrm{Re}(s) > \alpha$, it follows that $\alpha\{\hat{f}\} \leq \inf\{\omega : \omega > \omega\{\hat{f}\}\} = \omega\{\hat{f}\}$. □

Actually, the abscissa is characterized by the antiderivative of $|\hat{f}|$:

**Theorem 2.7** ([6, Theorem 1.4.3], cf. [94, Ch. II §2.4, §3.2])**.** *Let $\hat{f} \in L^1_{\mathrm{loc}}(\mathbb{R}_0^+)$ and*

$$F(t) = \int_0^t |\hat{f}(z)|\,\mathrm{d}z, \quad F_\infty = \begin{cases} \lim_{t\to\infty} F(t) & \text{if the limit exists,} \\ 0 & \text{otherwise.} \end{cases}$$

*Then*[2]

$$\alpha\{\hat{f}\} = \omega\{F(t) - F_\infty\}.$$

[2] Including the constant $F_\infty$ might seem unnecessary at first sight. However, $F(t)$ is a monotonically increasing function, so $\omega\{F\} \geq 0$. By including $F_\infty$, we enable negative values, too.

Laplace transforms have some nice properties. From the definition, it immediately follows that Laplace transforms are linear, i.e.,

$$\mathcal{L}\{c_1\hat{f}_1(t) + c_2\hat{f}_2(t)\}(s) = c_1\mathcal{L}\{\hat{f}_1\}(s) + c_2\mathcal{L}\{\hat{f}_2\}(s).$$

In addition, Laplace transforms are analytic functions and their derivatives are again Laplace transforms.

**Theorem 2.8** ([25, Theorem 6.1])**.** *Let* $\alpha\{\hat{f}\} < \infty$*. Then all derivatives of* $\mathcal{L}\{\hat{f}\}(s)$ *exist for* $\mathrm{Re}(s) > \alpha\{\hat{f}\}$ *and they are given by*

$$\frac{\mathrm{d}^n}{\mathrm{d}s^n}\mathcal{L}\{\hat{f}(t)\}(s) =: \mathcal{L}\{\hat{f}(t)\}^{[n]}(s) = (-1)^n\mathcal{L}\{t^n\hat{f}(t)\}(s), \quad n \in \mathbb{N}.$$

**Example 2.9.** We have

$$\mathcal{L}_t\left\{\frac{t^{\beta-1}}{\Gamma(\beta)}\exp(at)\right\}(s) = (s-a)^{-\beta}, \quad \mathrm{Re}(\beta) > 0, \quad \alpha\{\hat{f}\} = \mathrm{Re}(a),$$

see [25, Table of Laplace Transforms]. Here, $\Gamma(\beta)$ denotes the gamma function

$$\Gamma(\beta) = \int_0^\infty t^{\beta-1}\exp(-t)\,\mathrm{d}t, \quad \mathrm{Re}(\beta) > 0,$$

which is a generalization of the factorial, i.e., $\Gamma(n) = (n-1)!$ for $n \in \mathbb{N}$.

Before we discuss the other classes of functions that are relevant in this thesis (and their relation to Laplace transforms), we remark on alternative definitions of Laplace transforms.

**Remark 2.10.** Definitions other than Definition 2.1 have been considered in the literature. In [6, 25], Laplace transforms are defined as

$$\lim_{\omega\to\infty}\int_0^\omega \exp(-ts)\hat{f}(t)\,\mathrm{d}t,$$

i.e., as an *improper* Lebesgue integral. This allows for more functions $\hat{f}$ than Definition 2.1 but the additional limit complicates working with the transforms. In particular, the region of existence of $\hat{f}$ (now called the "region of convergence") and the one of $|\hat{f}|$ ("region of absolute convergence") do not necessarily coincide anymore, i.e., $\alpha\{\hat{f}\} \leq \alpha\{|\hat{f}|\}$. On the other hand, the two regions are identically equal, $\alpha\{\hat{f}\} \equiv \alpha\{|\hat{f}|\}$, in our case by the definition of Lebesgue integration. Note that our definition is the special case where the above limit converges to a finite value not only for $\hat{f}$ but also for $|\hat{f}|$ (see Lemma A.1 for more details). Thus, we can still use results from [6, 25] if we apply them to $|\hat{f}|$ instead of $\hat{f}$. Note also that proofs in [25] are often given in terms of improper Riemann integrals but—as mentioned on p. 11 in [25]—"the statements remain essentially unchanged".

Schilling et al. [80] use the proper integral

$$\int_0^\infty \exp(-ts)\,\mathrm{d}\mu(t),$$

where $\mu$ is a positive measure. This definition allows one to express every completely monotone function (see Definition 2.22) as a Laplace transform, see [80, Theorem 1.4]. On the other hand, it does not include all possible Laplace transforms of Definition 2.1 but only those with non-negative $\hat{f}$ (see Lemma A.4). Thus, their results hold for us (only) for non-negative $\hat{f}$ or for $|\hat{f}|$.

Relevant examples—such as those that we consider in Chapter 4—are usually contained in all three definitions. Thus, we use our conceptually simpler definition of a proper Lebesgue integral in this thesis: This way we can present the main ideas without cluttering the proofs with technical details. Nonetheless, we briefly discuss the relationship between the definitions and whether our main result (Corollary 4.8) holds for those, too, in Appendix A.

### 2.1.2. Rational functions

Another class of functions we consider is the class of rational functions.

**Definition 2.11.** Let $p$ and $q \not\equiv 0$ be polynomials. Then the function

$$r(s) = \frac{p(s)}{q(s)}, \quad s \in \mathbb{C}\backslash\{s : q(s) = 0\},$$

is called a *rational function*.

We might want to classify rational functions by the degrees $n$ and $m$ of the numerator polynomial and denominator polynomial. However, the simple example $r(s) = \frac{s^k}{s^k} = 1$ for $k \in \mathbb{N}$, $s \neq 0$ shows that these degrees are not unique for a given $r$.

**Definition 2.12.** Let $r$ be a rational function. Let $\tilde{p}$ and $\tilde{q}$ be polynomials of lowest degree $\tilde{n}$ and $\tilde{m}$, respectively, such that

$$\frac{\tilde{p}(s)}{\tilde{q}(s)} = r(s).$$

Then we call $(\tilde{n}, \tilde{m})$ the *degree* of $r$.

Working with rational functions is often simplified by decomposing them into a sum of simpler functions.

**Theorem 2.13** (e.g., [89, Theorem 23.1])**.** *Let $r(s) = \frac{p(s)}{q(s)}$ be a rational function of degree $(n, m)$ with $n, m > 0$. Then $r$ has the unique representation*

$$r(s) = p_0(s) + \sum_{j=1}^{k} \sum_{i=1}^{m_j} \frac{w_{j,i}}{(s - \tau_j)^i},$$

*where* $q(\tau_j) = 0$, $m = \sum_{j=1}^{k} m_j$ *and* $p_0$ *is a polynomial of degree* $n_0 \leq n$. *We call this representation* partial fraction expansion.

Several algorithms that compute the partial fraction expansion have been known for a long time, see, e.g., [66]. If the polynomial part $p_0$ vanishes, $p_0 \equiv 0$, rational functions are Laplace transforms (for large enough $\mathrm{Re}(s)$). This is often used for solving simple differential equations, see, e.g., [25, Chapter 15].

**Corollary 2.14.** *Let* $r$ *be a rational function. Then it can be written as*

$$r(s) = p_0(s) + \sum_{j=1}^{k} \sum_{i=1}^{m_j} \frac{w_{j,i}}{(i-1)!} \mathcal{L}_t\{t^{i-1} \exp(\tau_j t)\}(s)$$

*for* $\mathrm{Re}(s) > \max_{j=1,\dots,k} \mathrm{Re}(\tau_j)$.

*Proof.* Theorem 2.13 and Example 2.9. □

Rational matrix functions are usually not of interest by themselves but only as approximations of other functions. For instance, given a function $f$, the rational function $r$ that minimizes the maximum error in a region for a given degree $(n, m)$ is sometimes called the *rational minimax approximation* to $f$, see [89, Ch. 24]. Another popular choice of rational approximations is the class of *Padé approximations*, which we discuss now based on [22, 89].

**Definition 2.15.** Let $f(s)$ be analytic at $\xi$. Let $r_{n,m}$ be a rational function of degree $(n, m)$ such that

$$|f(s) - r_{n,m}(s)| = \mathcal{O}((s-\xi)^{n+m+1}),$$

i.e., the first $n+m+1$ terms of the Taylor expansions of $f$ and $r_{n,m}$ at $\xi$ coincide. Then we call $r_{n,m}$ the *Padé approximation of degree* $(n, m)$ *to* $f$.

If it is possible, $\xi$ is typically chosen to be 0. Otherwise, $\xi = 1$ is often chosen. Padé approximations of varying degrees are sometimes ordered in a table

$$\begin{array}{llllll} r_{0,0} & r_{0,1} & r_{0,2} & r_{0,3} & \cdots & \\ r_{1,0} & r_{1,1} & r_{1,2} & \cdots & & \\ r_{2,0} & r_{2,1} & \ddots & & & \\ r_{3,0} & \vdots & & & & \\ \vdots & & & & & \end{array} .$$

Because of this, the Padé approximations with degree $(n, n)$ are called *diagonal*. Similarly, a sequence $\{r_{k,0}, r_{k+1,0}, r_{k+1,1}, r_{k+2,1}, \dots\}$ of Padé approximations for $k \geq 0$ is called a *descending staircase* [22, Eq. (4.3.1)]. Note that the first column of the Padé table is just the sequence of the Taylor approximations of $f$. Moreover, the following holds:

**Lemma 2.16** ([22, Section 4.2])**.** *Let $r_{n,m}(s) = \frac{p_{n,m}(s)}{q_{n,m}(s)}$ be the Padé approximation of degree $(n,m)$ for $f(s)$. Then the Padé approximation of degree $(m,n)$ for $\frac{1}{f(s)}$ is given by $\frac{1}{r_{n,m}(s)} = \frac{q_{n,m}(s)}{p_{n,m}(s)}$.*

Padé approximations of different degrees may coincide. This can only occur if all approximations inside a square block

$$\begin{matrix} r_{n,m} & r_{n,m+1} & \cdots & r_{n,m+k} \\ r_{n+1,m} & r_{n+1,m+1} & \cdots & r_{n+1,m+k} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n+k,m} & r_{n+k,m+1} & \cdots & r_{n+k,m+k} \end{matrix}$$

coincide, however, see [22, Theorem 4.2.1].

### 2.1.3. Other classes of functions

#### Two-sided Laplace transforms

**Definition 2.17.** The *two-sided Laplace transform* of a function $\hat{f} \in L^1_{\mathrm{loc}}(\mathbb{R})$ is defined by the proper Lebesgue integral

$$\int_{-\infty}^{\infty} \exp(-ts)\hat{f}(t)\,\mathrm{d}t$$

whenever it has a finite value.

Compared to the Laplace transforms from Definition 2.1, two-sided Laplace transforms extend the domain of integration to negative numbers (see [25, Chapter 24]). To distinguish two-sided Laplace transforms from the Laplace transforms of Definition 2.1, we may call the latter *one-sided*. The two-sided transforms can be interpreted as the generalization of the Fourier transform

$$\int_{-\infty}^{\infty} \exp(-ti\omega)\hat{f}(t)\,\mathrm{d}t, \quad \omega \in \mathbb{R},$$

to complex values for $\omega$. From basic properties of Lebesgue integrals, it follows

$$\begin{aligned} \int_{-\infty}^{\infty} \exp(-ts)\hat{f}(t)\,\mathrm{d}t &= \int_{0}^{\infty} \exp(-ts)\hat{f}(t)\,\mathrm{d}t + \int_{-\infty}^{0} \exp(-ts)\hat{f}(t)\,\mathrm{d}t \\ &= \mathcal{L}\{\hat{f}(t)\}(s) + \mathcal{L}\{\hat{f}(-t)\}(-s) \end{aligned}$$

whenever the integral on the left or the two on the right are finite (cf. [77, Theorem 1.32]). Thus, if we want to evaluate two-sided Laplace transforms, we can fall back to algorithms for one-sided ones. However, both $\mathcal{L}\{\hat{f}(t)\}(s)$ and $\mathcal{L}\{\hat{f}(-t)\}(-s)$ need to exist for the same value $s$. The exact region of existence is thus the strip $\alpha\{\hat{f}(t)\} < \mathrm{Re}(s) < -\alpha\{\hat{f}(-t)\}$, possibly including $\mathrm{Re}(s) = \alpha\{\hat{f}\}$ or $\mathrm{Re}(s) = -\alpha\{\hat{f}(-t)\}$. Note that the one-sided Laplace transforms are the subclass of the two-sided ones with $\hat{f}(t) = 0$ for $t < 0$.

### Entire functions

**Definition 2.18.** A function $f$ is *entire* if it is analytic on the whole complex plane. An entire function is *of order one* if

$$\limsup_{r\to\infty} \frac{\log\log M(r)}{\log r} = 1, \quad M(r) = \max_{|s|=r} |f(s)|.$$

See [15] for more information about entire functions.

**Example 2.19.** The most prominent example of an entire function is the exponential function $f(s) = \exp(as)$ with $a \in \mathbb{C}$. It is easily verified that it is also of order one.

Laplace transforms are analytic in their region of existence. Consequently, if $\alpha\{\hat{f}\} = -\infty$, then $\mathcal{L}\{\hat{f}\}$ is an entire function. One way to construct such functions is by restricting the region of integration to a finite interval.

**Lemma 2.20.** *Let $T \geq 0$ be finite and $\hat{f} \in L^1([0,T])$. Then*

$$\int_0^T \exp(-ts)\hat{f}(t)\,\mathrm{d}t < \infty$$

*for every value of $s \in \mathbb{C}$.*

*Proof.* This immediately follows from

$$\int_0^T \exp(-t\,\mathrm{Re}(s))|\hat{f}(t)|\,\mathrm{d}t \leq c_s \int_0^T |\hat{f}(t)|\,\mathrm{d}t, \quad c_s = \begin{cases} \exp(-T\,\mathrm{Re}(s)), & \mathrm{Re}(s) < 0, \\ 1, & \mathrm{Re}(s) \geq 0. \end{cases}$$ □

**Corollary 2.21** ([25, Theorem 6.2])**.** *Let $T \geq 0$ be finite and $\hat{f} \in L^1([0,T])$. Denote by $\chi_{(0,T)}(t)$ the characteristic function of the interval $(0,T)$. A Laplace transform of the form*

$$\mathcal{L}\{\hat{f}(t)\chi_{(0,T)}(t)\}(s) = \int_0^T \exp(-ts)\hat{f}(t)\,\mathrm{d}t$$

*is an entire function.*

We are interested in entire functions when we do polynomial interpolation: If one interpolates an entire function such that the maximum error is minimized, one can show that this error converges superlinearly to 0 when increasing the degree of the polynomial, see, e.g., [37]. This has been used when analyzing the convergence of the (restarted) Arnoldi method, see Theorem 2.70. We give a variation of this in Section 4.4.1.

### Completely monotone functions

**Definition 2.22.** A function $f : \mathbb{R}^+ \to \mathbb{R}_0^+$ is called *completely monotone* if all its derivatives $f^{[n]}(s)$ exist for $s \in \mathbb{R}^+$ and

$$(-1)^n f^{[n]}(s) \geq 0 \quad \text{for all } s \in \mathbb{R}^+, n \in \mathbb{N}.$$

See [80, Chapter 1] for more information. It immediately follows from Theorem 2.8 that many Laplace transforms are completely monotone.

**Lemma 2.23** ([25, Theorem 31.9], [80, Theorem 1.4])**.** *Let $\hat{f} \geq 0$ and $\alpha\{\hat{f}\} \leq 0$. Then $\mathcal{L}\{\hat{f}\}(s)$ (restricted to $s \in \mathbb{R}^+$) is a completely monotone function.*

**Example 2.24.** The function $f(s) = s^{-1} = \mathcal{L}\{1\}(s)$ is completely monotone.

An important property of completely monotone functions is their closure under multiplication.

**Lemma 2.25** ([80, Corollary 1.6])**.** *Let $f(s)$ and $g(s)$ be completely monotone functions. Then the product $f(s)g(s)$ is also completely monotone.*

Integrals of a completely monotone function can be bounded from below and from above by Gauß quadrature rules, see [46]. Based on this, one can compute a posteriori error bounds for the restarted Arnoldi method for $f(A)\boldsymbol{b}$, see Theorem 2.72.

### Stieltjes functions

**Definition 2.26.** A function $f : \mathbb{C}\backslash\mathbb{R}_0^- \to \mathbb{C}$ defined by

$$f(s) = b + \int_0^\infty \frac{\rho(t)}{t+s}\,\mathrm{d}t$$

with $b \geq 0$ and $\rho \geq 0$ such that

$$\int_0^\infty \frac{\rho(t)}{t+1}\,\mathrm{d}t < \infty$$

is called a *Stieltjes function*.[3]

Since $(t+s)^{-1} = \mathcal{L}_\tau\{\exp(-t\tau)\}(s)$ for $\mathrm{Re}(s) > t$, Stieltjes functions can be interpreted as iterated Laplace transforms:

**Lemma 2.27** ([80, Theorem 2.2])**.** *Every Stieltjes function $f$ has a restriction to $\mathrm{Re}(s) > 0$ of the form*

$$f(s) = b + \mathcal{L}\{\mathcal{L}\{\rho\}(t)\}(s).$$

[3] Stieltjes functions are often defined more generally, see, e.g., [12, 80]. This is done by using other measures than $\rho(t)\,\mathrm{d}t$. As we restrict ourselves to the measure $\hat{f}(t)\,\mathrm{d}t$ for Laplace transforms in a similar way (cf. Remark 2.10), we use this simpler definition here.

Note that if $g = \mathcal{L}\{\rho\}$ is completely monotone, then $\mathcal{L}\{g\}$ is also completely monotone. From Lemma 2.23, we immediately arrive at the following corollary:

**Corollary 2.28** ([80, Theorem 2.2])**.** *A Stieltjes function (restricted to $\mathbb{R}^+$) is a completely monotone function.*

We also have the following result regarding Padé approximations:

**Lemma 2.29** ([22, Theorem 4.2.3])**.** *Let $r_{n,m}$ and $r_{k,\ell}$ be two Padé approximations to a Stieltjes function $f$. Then $r_{n,m} \equiv r_{k,\ell}$ implies $(n,m) = (k,\ell)$. In other words, Padé approximations to Stieltjes functions are pairwise distinct.*

**Example 2.30.** The function $f(s) = s^{-1/2}$ is a Stieltjes function with $b = 0$ and $\rho(t) = \pi^{-1}t^{-1/2}$, [13, §14.15].

The restarted Arnoldi method (Section 2.4.2) has been described in [39, 40, 41, 49, 81] for Stieltjes functions. In Chapter 4, we extend their method to Laplace transforms.

### Bernstein functions

**Definition 2.31** ([80, Definition 3.1])**.** A function $f : \mathbb{R}^+ \to \mathbb{R}_0^+$ is called a *Bernstein function* if all its derivatives $f^{[n]}(s)$ exist in $\mathbb{R}^+$ and

$$(-1)^{n-1} f^{[n]}(s) \geq 0, \quad n \in \mathbb{N}.$$

In other words, its derivative is completely monotone.

Bernstein functions have the following integral representation.

**Theorem 2.32** ([80, Theorem 3.2])**.** *A function $f : \mathbb{R}^+ \to \mathbb{R}_0^+$ is a Bernstein function if and only if it has the* Lévy-Khintchine representation*, i.e.,*

$$f(s) = a + bs + \int_0^\infty (1 - \exp(-ts))\, \mathrm{d}\mu(t),$$

*where $a, b \geq 0$ and $\mu$ is a measure such that*

$$\int_0^\infty \min(1,t)\, \mathrm{d}\mu(t) < \infty.$$

This integral representation shows that Bernstein functions can be extended to the right half-plane:

**Corollary 2.33** ([80, Proposition 3.6])**.** *Every Bernstein function $f$ has a holomorphic extension to* $\mathrm{Re}(s) > 0$*.*

In what follows, we do not differentiate strictly between the original function $f$ and its extension. We also consider only the following subclass of Bernstein functions.

**Definition 2.34.** A Bernstein function $f$ is called *complete* if the measure $\mu$ in the Lévy-Khintchine representation has a completely monotone derivative with respect to the Lebesgue measure, i.e.,

$$f(s) = a + bs + \int_0^\infty (1 - \exp(-ts))\mu'(t)\,\mathrm{d}t.$$

Complete Bernstein functions are closely related to Laplace transforms and Stieltjes functions:

**Lemma 2.35.** *Let $f$ be a complete Bernstein function. Then its derivative $f'$ is essentially a Laplace transform, more precisely,*

$$f'(s) = b + \int_0^\infty \exp(-ts)t\mu'(t)\,\mathrm{d}t = b + \mathcal{L}\{t\mu'(t)\}(s).$$

*Proof.* In the proof of Theorem 3.2 in [80], it is shown that

$$f'(s) = b + \int_0^\infty \exp(-ts)t\,\mathrm{d}\mu(t)$$

if $f$ is a Bernstein function with $b$ and $\mu$ as in Theorem 2.32. □

**Theorem 2.36** ([80, Theorem 6.2])**.** *Let $f$ be a complete Bernstein function. If $a = 0$ and if $\mu'$ can be represented by a Laplace transform $\mu'(t) = \mathcal{L}\{g\}(t)$, then $s^{-1}f(s)$ is a Stieltjes function (restricted to $\mathrm{Re}(s) > 0$).*[4]

*Conversely, if $f$ is a Stieltjes function, then $sf(s)$ (restricted to $\mathrm{Re}(s) > 0$) is a complete Bernstein function.*

**Example 2.37.** The function $f(s) = \sqrt{s}$ is a complete Bernstein function with $a, b = 0$ and $\mu'(t) = (2\sqrt{\pi})^{-1}t^{-3/2}$, [80, Section 16.2, No. 1].

It turns out that complete Bernstein functions can be treated exactly as Laplace transforms in the restarted Arnoldi method, see Section 4.1.2.

## 2.2. Continued fractions

In Chapter 3, we introduce a new method for evaluating rational matrix functions. For this, we need to represent rational functions as continued fractions. Before we can do this, however, we need to define continued fractions first without reference to functions. This section presents well-known results about continued fractions that can be found, e.g., in [22, 54]. In parts, we use a similar description as in [CF].

We use the extended complex numbers $\mathbb{C}_\infty = \mathbb{C} \cup \{\infty\}$ with the usual arithmetic conventions.

[4] Using a more general definition for Stieltjes functions, this holds for *all* complete Bernstein functions.

**Definition 2.38.** Given two (finite or infinite) sequences $\{c_i\}_{i=1}^{\alpha}$ and $\{b_i\}_{i=0}^{\alpha}$ with $c_i, b_i \in \mathbb{C}$ and $\alpha \in \mathbb{N} \cup \{\infty\}$, we define the functions $z_i : \mathbb{C}_\infty \to \mathbb{C}_\infty$ for $i = 0, 1, \ldots$ to be

$$z_0(s) = b_0 + s, \quad z_j(s) = \frac{c_j}{b_j + s} \text{ for } j \geq 1,$$

and denote their compositions (with 0 as starting argument) by

$$g_i \coloneqq z_0 \circ z_1 \circ \cdots \circ z_i(0) \in \mathbb{C}_\infty.$$

We call

$$g = \begin{cases} g_\alpha & \text{if } \alpha \in \mathbb{N}, \\ \lim_{i\to\infty} g_i & \text{if } \alpha = \infty \end{cases}$$

the *formal continued fraction* of $(\{c_i\}_{i=1}^{\alpha}, \{b_i\}_{i=0}^{\alpha})$, which we write as

$$g =: b_0 + \operatorname*{K}_{i=1}^{\alpha} \left(\frac{c_i}{b_i}\right).$$

We further call the elements $c_i$, $b_i$ and $g_i$ the $i$th *partial numerator*, *partial denominator* and *approximant*, respectively.

If $\alpha \in \mathbb{N}$, we may call $g$ *finite*; if $\alpha = \infty$, we may call $g$ *infinite*.

While we allow division by 0 by working within $\mathbb{C}_\infty$, the approximants (and in turn the formal continued fractions) are only defined if the term $\frac{0}{0}$ does not occur. It is common in number theory to assume that $c_i \neq 0$ for all $i$ to avoid this (e.g., [54, §12.1]). It will be useful later on to include cases where $c_i$ potentially vanishes, so we do not assume this. Note, however, that if $g_j$ is defined and $c_i = 0$ for some $i < j$, then $g_j = g_{i-1}$.

**Definition 2.39.** Let $g$ be the formal continued fraction of $\{c_i\}_{i=1}^{\alpha}$ and $\{b_i\}_{i=0}^{\alpha}$. If the term $\frac{0}{0}$ appears in an approximant $g_i$, we call $g_i$ *undefined*, otherwise *defined*.

- If $g$ is finite and $g = g_\alpha$ is defined, then we call $g$ a (finite) *continued fraction*.
- If $g$ is infinite and has only a finite number of undefined approximants, then we call $g$ an (infinite) *continued fraction* and take the limit only over all defined approximants.

While Definitions 2.38 and 2.39 seem somewhat abstract, they enable the simple representations

$$g = b_0 + \operatorname*{K}_{i=1}^{\infty} \left(\frac{c_i}{b_i}\right) = b_0 + \cfrac{c_1}{b_1 + \cfrac{c_2}{b_2 + \ldots}} \in \mathbb{C}_\infty$$

for an infinite continued fraction $g$ and

$$g_m = b_0 + \operatorname*{\mathrm{K}}_{i=1}^{m}\left(\frac{c_i}{b_i}\right) = b_0 + \cfrac{c_1}{b_1 + \cfrac{c_2}{b_2 + \quad \ddots \quad + \cfrac{c_m}{b_m}}} \in \mathbb{C}_\infty$$

for both an $m$th approximant and a finite continued fraction.

**Lemma 2.40** ([54, Corollary 12.1b])**.** *Let $g$ be a continued fraction. Then the approximants are simple fractions $g_m = \frac{p_m}{q_m}$, where the numerator and denominator fulfill the recurrence relation*

$$\begin{bmatrix} p_{-1} \\ q_{-1} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} p_0 \\ q_0 \end{bmatrix} = \begin{bmatrix} b_0 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} p_m \\ q_m \end{bmatrix} = b_m \begin{bmatrix} p_{m-1} \\ q_{m-1} \end{bmatrix} + c_m \begin{bmatrix} p_{m-2} \\ q_{m-2} \end{bmatrix} \text{ for } m \geq 1.$$

**Corollary 2.41.** *Lemma 2.40 also holds if $g_m$ is undefined in the sense that the recursion yields $p_m = 0$ and $q_m = 0$ in that case.*

*Proof.* If $g_m$ is undefined because $b_m = c_m = 0$, then the statement immediately follows from the recursion. Otherwise, let

$$t_i \coloneqq \cfrac{c_{i+1}}{b_{i+1} + \cfrac{c_{i+2}}{b_{i+2} + \quad \ddots \quad + \cfrac{c_m}{b_m}}} = z_{i+1} \circ \cdots \circ z_m(0),$$

where $i < m$ is chosen as large as possible such that $b_i + t_i = c_i = 0$. Then $t_i$ is defined and we can write

$$g_m = b_0 + \cfrac{c_1}{b_1 + \cfrac{c_2}{b_2 + \quad \ddots \quad + \cfrac{c_i}{b_i + t_i}}}.$$

This is again the case where the last fraction is $\frac{0}{0}$, so we have

$$\begin{bmatrix} p_m \\ q_m \end{bmatrix} = (b_i + t_i) \begin{bmatrix} p_{i-1} \\ q_{i-1} \end{bmatrix} + c_i \begin{bmatrix} p_{i-2} \\ q_{i-2} \end{bmatrix} = 0. \qquad \square$$

We mention some results that are useful later on: Another way of evaluating a finite continued fraction is by solving a linear system.

**Theorem 2.42** ([71, Theorem 1], cf. [CF, Theorem 3.1])**.** *Let* $g_m = b_0 + \mathrm{K}_{i=1}^{m} \left(\frac{c_i}{b_i}\right) \neq \infty$ *be a finite continued fraction. If the entries of the tridiagonal matrix*

$$T_m = \begin{bmatrix} \beta_1 & \gamma_2 & & & \\ \alpha_2 & \beta_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \beta_{m-1} & \gamma_m \\ & & & \alpha_m & \beta_m \end{bmatrix} \in \mathbb{C}^{m\times m}$$

*fulfill*

$$\begin{aligned} \beta_i &= b_i, \quad i = 1, \dots, m, \\ -\alpha_i\gamma_i &= c_i, \quad i = 2, \dots, m, \end{aligned}$$

*then* $T_m$ *is non-singular and*

$$g_m = b_0 + c_1 (T_m^{-1})_{1,1} = b_0 + c_1 \boldsymbol{e}_1^{\mathsf{T}} T_m^{-1} \boldsymbol{e}_1.$$

**Remark 2.43.** We have $g_m = \infty$ if and only if $T_m$ is singular, see [CF, Remark 3.2]. If we assign $\boldsymbol{e}_1^{\mathsf{T}} T_m^{-1} \boldsymbol{e}_1 = \infty$ for a singular matrix $T_m$, then the above equation still holds for $g_m = \infty$.

There is a way to multiply the partial numerators and partial denominators by constants without changing the approximants.

**Lemma 2.44** ([54, Eq. (12.1-11)])**.** *Let*

$$g = b_0 + \operatorname*{K}_{i=1}^{\alpha} \left(\frac{c_i}{b_i}\right)$$

*be a continued fraction. Then its approximants coincide with the ones of*

$$\tilde{g} = b_0 + \operatorname*{K}_{i=1}^{\alpha} \left(\frac{d_{i-1} d_i c_i}{d_i b_i}\right)$$

*with* $d_0 = 1$ *and* $d_i \neq 0$*, i.e., we have* $g_i = \tilde{g}_i$ *and* $g = \tilde{g}$*.*

Analogously, we can transform the matrix $T_m$ in Theorem 2.42:

**Corollary 2.45** ([CF, Corollary 3.3])**.** *Theorem 2.42 still holds if* $T_m$ *is multiplied by any two non-singular matrices from the left and the right as long as* $\boldsymbol{e}_1$ *is a right and left eigenvector, respectively, of these matrices with reciprocal eigenvalue:*

$$\left.\begin{aligned} H_\ell^{-1} \boldsymbol{e}_1 &= \lambda \boldsymbol{e}_1 \\ \boldsymbol{e}_1^{\mathsf{T}} H_r^{-1} &= \lambda^{-1} \boldsymbol{e}_1^{\mathsf{T}} \end{aligned}\right\} \implies g_m = b_0 + \boldsymbol{e}_1^{\mathsf{T}} T_m^{-1} \boldsymbol{e}_1 = b_0 + \boldsymbol{e}_1^{\mathsf{T}} (H_\ell T_m H_r)^{-1} \boldsymbol{e}_1.$$

*In particular, using diagonal matrices* $D = \operatorname{diag}(1, d_2, \dots, d_m)$ *for* $H_\ell$ *and* $H_r$ *is equivalent to expanding the continued fraction as in Lemma 2.44.*

It will turn out useful to reduce the length of a continued fraction.

**Lemma 2.46** ([22, Eq. (1.5.3)])**.** *Let $\alpha$ be even and*

$$g = b_0 + \operatorname*{K}_{i=1}^{\alpha} \left( \frac{c_i}{b_i} \right)$$

*be a continued fraction. Define the continued fraction*

$$\tilde{g} = b_0 + \operatorname*{K}_{i=1}^{\alpha/2} \left( \frac{a_i}{d_i} \right)$$

*with $a_1 = c_1 b_2$, $d_1 = c_2 + b_1 b_2$ and*

$$a_i = -\frac{c_{2i-2} c_{2i-1} b_{2i}}{b_{2i-2}}, \quad d_i = c_{2i} + b_{2i-1} b_{2i} + \frac{c_{2i-1} b_{2i}}{b_{2i-2}}, \quad i \geq 2.$$

*Then every approximant of $\tilde{g}$ coincides with every other approximant of $g$, i.e., $g_{2i} = \tilde{g}_i$ for $i \geq 0$. We call $\tilde{g}$ the* contraction *of $g$.*

The connection between continued fractions and rational functions becomes apparent if we replace the partial numerators and denominators $c_i$ and $b_i$ by polynomials $c_i(s)$ and $b_i(s)$. Then $p_i(s)$ and $q_i(s)$ in the recursion of Lemma 2.40 are polynomials, too. This means the approximants are the rational functions $g_i(s) = \frac{p_i(s)}{q_i(s)}$. Now we also see why we did not demand that $c_i \neq 0$ in Definitions 2.38 and 2.39: For non-constant polynomials $c_i(s)$, we can find at least one value of $s$ such that $c_i(s) = 0$ but an approximant $g_i(s)$ is not necessarily undefined in this case.

If the approximant $g_i(s)$ is undefined for some value $s$, however, then $p_i(s), q_i(s) = 0$ by Corollary 2.41. We can easily avoid this situation by considering the rational functions $g_i(s)$ only for values of $s$ such that $q_i(s) \neq 0$. This way, we also have $g_i(s) \in \mathbb{C}$ for all remaining $s$. Thus, a pair of polynomial sequences $(\{c_i(s)\}_{i=1}^{\alpha}, \{b_i(s)\}_{i=0}^{\alpha})$ interpreted as a continued fraction generates a sequence of rational functions $\{r_i(s) = g_i(s)\}_{i=0}^{\alpha}$.

One might wonder now which continued fractions yield interesting rational functions. We briefly discuss this here (based on [CF, 22]). We are interested in rational functions as approximations to other functions. In Section 2.1.2, we introduced the Padé approximations as examples. It turns out they are closely connected to continued fractions.

**Definition 2.47.** A continued fraction of the form

$$b_0 + \operatorname*{K}_{i=1}^{\infty} \left( \frac{c_i s^{n_i}}{1} \right), \quad b_0 \in \mathbb{C},\ c_i \in \mathbb{C}\backslash\{0\},\ n_i \in \mathbb{N},$$

is called a *C-fraction*. If $n_i = 1$ for all $i$, then the C-fraction is called *regular*.

**Theorem 2.48** ([22, Theorem 4.3.1])**.** *Let $S = \{r_{k,0}, r_{k+1,0}, r_{k+1,1}, r_{k+2,1}, \dots\}$, $k \geq 0$, be a descending staircase of Padé approximations to a function $f$. If every three consecutive elements of $S$ are distinct, then there exists a C-fraction*

$$r_{k,0}(s) + s^k \operatorname*{\mathrm{K}}_{i=1}^{\infty} \left( \frac{c_i s}{1} \right), \quad c_i \neq 0,$$

*such that its $m$th approximant is the $(m+1)$st element of $S$ for all $m \geq 0$.*

Applying the contraction from Lemma 2.46, we immediately obtain the following modification.

**Corollary 2.49.** *Let $r_{k,k}(s)$ denote the diagonal Padé approximations to a function $f$. If $r_{k,k} \not\equiv r_{k+1,k+1}$ for all $k$, then there exists a continued fraction*

$$r_{0,0} + \cfrac{c_1 s}{1 + c_2 s + \mathrm{K}_{i=2}^{\infty} \left( \frac{-c_{2i-2} c_{2i-1} s^2}{1 + (c_{2i} + c_{2i-1}) s} \right)}, \quad c_i \neq 0,$$

*such that its $m$th approximant is the diagonal Padé approximation $r_{m,m}(s)$.*

The condition $r_{k,k} \not\equiv r_{k+1,k+1}$ is fulfilled, for example, if $f$ is a Stieltjes function, see Lemma 2.29.

Of course, other kinds of continued fractions also yield approximations to functions. Describing them in detail would go beyond the scope of this thesis. We refer to [22, Part III] for many examples of continued fractions whose approximants $g_m(s)$ converge to functions for $m \to \infty$.

We do not cover how to construct continued fractions in detail and only give some pointers: The coefficients of C-fractions can be found from Taylor series using the *qd algorithm* [22, Section 6.1]. So-called *Thiele fractions* [22, Section 6.8] allow the construction of interpolating rational functions. If a rational function $r(s) = \frac{p(s)}{q(s)}$ is already given as a fraction of two polynomials, then one continued fraction is of course just $g_1(s) = 0 + \frac{p(s)}{q(s)+0}$. One might, however, want to find longer continued fractions where the partial numerators and denominators have lower degrees than $p$ and $q$. One way to achieve this is to apply the *Euclidean algorithm* on $p$ and $q$ ([53, §6.3 III.]). Here, we start with polynomial long division such that

$$\frac{p(s)}{q(s)} = b_0(s) + \frac{a(s)}{q(s)} = b_0(s) + \frac{1}{\frac{q(s)}{a(s)}},$$

where $a(s)$ is a polynomial of degree smaller than that of $q(s)$. Iterating on the resulting rational functions $\frac{q(s)}{a(s)}$, we obtain a continued fraction. Specifically, if $r(s)$ has degree $(n, n')$, then this results in

$$r(s) = \frac{p(s)}{q(s)} = b_0(s) + \operatorname*{\mathrm{K}}_{i=1}^{m} \left( \frac{1}{b_i(s)} \right), \tag{2.2}$$

with $m \leq n'$, see [62].

## 2.3. Definition of matrix functions

Throughout this work, we are interested in computing the action of a matrix function on a vector, i.e.,

$$f(A)\boldsymbol{b},$$

where $A \in \mathbb{C}^{n\times n}$ and $\boldsymbol{b} \in \mathbb{C}^n$. Before we discuss numerical methods in the next section, we need to define $f(A)$ first. This section states the common (and for analytic functions equivalent) definitions of the matrix function $f(A)$ as given, e.g., in [56, Section 1.2].

Let us start by recalling the Jordan canonical form.

**Definition 2.50.** A matrix

$$J(\mu, m) = \mu I_m + S_m \in \mathbb{C}^{m\times m}$$

with

$$\mu \in \mathbb{C}, \quad S_m = \begin{bmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & 0 \end{bmatrix} \in \mathbb{C}^{m\times m}$$

is called a *Jordan block*.

It is well known that every matrix $A \in \mathbb{C}^{n\times n}$ has a *Jordan canonical form*, i.e., it can be written as

$$A = ZJZ^{-1}, \quad J = \bigoplus_{j=1}^{\ell} J(\mu_j, m_j) \in \mathbb{C}^{n\times n}, \quad Z \in \mathbb{C}^{n\times n},$$

where $\mu_j \in \operatorname{spec}(A)$ are eigenvalues of $A$ and $\sum_{j=1}^{\ell} m_j = n$. The symbol $\hat{\oplus}$ denotes the direct sum. Let us denote the pairwise distinct eigenvalues of $A$ by $\lambda_1, \ldots, \lambda_s$. The index $n_k$ of an eigenvalue $\lambda_k$ is the size of the largest Jordan block containing it, i.e.,

$$n_k = \max\{m_j : \mu_j = \lambda_k,\ j = 1, \ldots, \ell\}, \quad k = 1, \ldots, s.$$

**Definition 2.51.** We call a function $f$ *defined on the spectrum of the matrix* $A$ if the values

$$f^{[i]}(\lambda_k), \quad i = 0, \ldots, n_k - 1, \quad k = 1, \ldots, s$$

exist.

Definition 2.51 informs us about the conditions $f$ has to fulfill such that $f(A)$ is defined. We see this in our first definition of $f(A)$:

**Definition 2.52.** Let $f$ be defined on the spectrum of $A = ZJZ^{-1}$. Then

$$f(A) \coloneqq Zf(J)Z^{-1} = Z \overbrace{\bigoplus_{j=1}}^{\ell} f(J(\mu_j, m_j))Z^{-1} \text{ and}$$

$$f(J(\mu_j, m_j)) \coloneqq \begin{bmatrix} f(\mu_j) & f^{[1]}(\mu_j) & \dots & \frac{f^{[m_j-1]}(\mu_j)}{(m_j-1)!} \\ & f(\mu_j) & \ddots & \vdots \\ & & \ddots & f^{[1]}(\mu_j) \\ & & & f(\mu_j) \end{bmatrix}.$$

It immediately follows that matrix functions commute, i.e., $f(A)g(A) = g(A)f(A)$. The Jordan canonical form is rarely used in computations, however, as it can be very sensitive to perturbations. Though in the special case that $A$ is normal, i.e., unitarily diagonalizable, it follows from Definition 2.52 that $f(A) = Z \operatorname{diag}(f(\mu_1), \dots, f(\mu_n))Z^{\mathsf{H}}$. Thus, in this case, $f(A)$ and $f(A)\boldsymbol{b}$ can be computed by using the well-conditioned eigendecomposition of $A$. Typically, the computational burden scales like $\mathcal{O}(n^3)$ so this approach is only feasible for small matrices. We use this direct computation of $f(A)$ when $A$ is a small Hermitian matrix in Section 4.2.3.

Another way of defining $f(A)$ is by expressing it as a matrix polynomial. While they are, strictly speaking, a class of matrix functions, their definition immediately follows from the basic operations[5] already defined on matrices.

**Definition 2.53.** Let $f$ be a function defined on the spectrum of $A$. A function $p$ that fulfills

$$p^{[i]}(\lambda_k) = f^{[i]}(\lambda_k), \quad i = 0, \dots, n_k - 1, \quad k = 1, \dots, s,$$

is said to *interpolate $f$ at the spectrum of $A$.*

**Definition 2.54.** Let $f$ be a function defined on the spectrum of $A$. Further, let $p$ be the unique polynomial of degree less than $\sum_{k=1}^{s} n_k$ that interpolates $f$ at the spectrum of $A$. Then $p$ is said to be the *Hermite interpolating polynomial* and

$$f(A) \coloneqq p(A).$$

**Theorem 2.55** ([56, Theorem 1.12])**.** *Definition 2.52 and Definition 2.54 are equivalent.*

While Definition 2.54 might seem numerically more practical than Definition 2.52 at first sight, note that the interpolating polynomial $p$ depends not only on $f$ but also on the eigenvalues of $A$. Moreover, this approach would require $\mathcal{O}(n^4)$ floating point operations ($\mathcal{O}(n)$ matrix-matrix multiplications each of which costs $\mathcal{O}(n^3)$) and is numerically unstable, see [56, Section 4.8] and [43].

[5] Namely addition, scalar multiplication, matrix-matrix multiplication.

Expressing $f(A)$ as an interpolating polynomial is still helpful for the theory discussed in Sections 2.4 and 4.4. Note that *any* polynomial that interpolates $f$ at the spectrum of $A$ yields $f(A) = p(A)$ (see [56, Theorem 1.3, Remark 1.5]). We can construct a polynomial, e.g., without knowing the indices $n_k$:

**Lemma 2.56** ([56, Remarks 1.5 and 1.6])**.** *Let $\mu_1, \dots, \mu_n$ denote the eigenvalues of $A$ including algebraic multiplicity. Let*

$$\mathbb{\Delta}_{[\mu_1,\dots,\mu_j]}\{f\}$$

*be the divided difference of $f$ with nodes $\mu_1, \dots, \mu_j$. Then the polynomial*

$$q(s) = \sum_{j=1}^{n} \mathbb{\Delta}_{[\mu_1,\dots,\mu_j]}\{f\} \cdot (s-\mu_1)\dots(s-\mu_{j-1})$$

*interpolates $f$ at the spectrum of $A$ and $q(A) = f(A)$.*

*If, in addition, the algebraic multiplicities coincide with the indices for all eigenvalues, $q$ is the Hermite interpolating polynomial from Definition 2.54.*

*Proof.* Let $m(\lambda_k)$ denote the algebraic multiplicity of the pair-wise distinct eigenvalues $\lambda_k$. By construction of $q$ by a Newton form, we have

$$q^{[i]}(\lambda_k) = f^{[i]}(\lambda_k), \quad i = 0, \dots, m(\lambda_k) - 1, \quad k = 1, \dots, s,$$

see, e.g., [16, Eq. (7)]. As the algebraic multiplicity is a bound for the index, i.e.,

$$n_k \leq \sum_{\substack{j=1 \\ \mu_j = \lambda_k}}^{\ell} m_j = m(\lambda_k),$$

$q$ interpolates $f$ at the spectrum of $A$. It now follows from [56, Theorem 1.3] that $q(A) = p(A) = f(A)$.

For the second statement, we have the hypothesis $n_k = m(\lambda_k)$ and thus

$$\sum_{k=1}^{s} n_k = \sum_{k=1}^{s} m(\lambda_k) = n.$$

On one hand, the Hermite interpolating polynomial $p$ is unique and of degree less than $\sum_{k=1}^{s} n_k$; on the other hand, $q$ has degree $n-1$. It follows that $q = p$. □

The previous two definitions solely assumed $f$ is defined on the spectrum of $A$. Here we assume that $f$ is analytic. Such functions can be represented as a Cauchy integral, which allows for the third and last definition:

**Definition 2.57.** Let $f$ be analytic in and on a closed contour $\Gamma$ enclosing $\operatorname{spec}(A)$. Then

$$f(A) \coloneqq \frac{1}{2\pi i} \int_{\Gamma} (sI - A)^{-1} f(s)\, \mathrm{d}s.$$

**Theorem 2.58** ([56, Theorem 1.12])**.** *If $f$ is analytic, then Definition 2.57 is equivalent to Definitions 2.52 and 2.54.*

The functions we consider in this thesis are all analytic, so Definition 2.57 is applicable and we use it theoretically in Section 4.4.1. Numerically, however, one would need to find a suitable contour and quadrature rule, which depend on both $f$ and $A$. For example, [23] discusses the case that $\Gamma$ is a simple circle and the repeated trapezoidal rule is used. See [52] for a more elaborate example.

## 2.4. Krylov subspace methods for general matrix functions

In this section, we consider approximations based on Krylov subspaces. Our description is guided by [78, 81]. We first define the subspace.

**Definition 2.59.** The $m$th *Krylov subspace* of $A \in \mathbb{C}^{n\times n}$ and $\boldsymbol{b} \in \mathbb{C}^n$ is given by

$$\mathcal{K}_m(A, \boldsymbol{b}) \coloneqq \operatorname{span}(\boldsymbol{b}, A\boldsymbol{b}, A^2\boldsymbol{b}, \ldots, A^{m-1}\boldsymbol{b}) = \{p(A)\boldsymbol{b} : p \in \mathcal{P}_{m-1}\},$$

where $\mathcal{P}_{m-1}$ is the set of all polynomials of degree at most $m-1$.

To evaluate $f(A)\boldsymbol{b}$, we project $f(A)\boldsymbol{b}$ onto the subspace $\mathcal{K}_m(A, \boldsymbol{b})$ for increasing $m$ until some convergence criterion is fulfilled.

### 2.4.1. The Arnoldi approximation

To construct an approximation $\boldsymbol{f}_m$ to $f(A)\boldsymbol{b}$, we need to construct a basis for $\mathcal{K}_m(A, \boldsymbol{b})$ first. For numerical stability, we want an orthonormal basis and this is usually obtained using the Arnoldi process, which is described in Algorithm 1. The idea is to start with $\boldsymbol{v}_1 = \frac{1}{\|\boldsymbol{b}\|_2}\boldsymbol{b}$ and iteratively construct new basis vectors by orthogonalizing $A\boldsymbol{v}_j$ against the previous basis vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_j$. In Algorithm 1, we use Modified Gram-Schmidt for orthogonalization. We do not discuss other variants and refer instead to [78, Section 6.3.2].

The Arnoldi process yields the matrix $V_m \in \mathbb{C}^{n\times m}$, whose columns contain the orthonormal basis vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$ for $\mathcal{K}_m(A, \boldsymbol{b})$, and an upper Hessenberg matrix $H_m = [h_{i,j}] \in \mathbb{C}^{m\times m}$. They fulfill the *Arnoldi relation* [78, Eq. (6.6)]

$$AV_m = V_m H_m + h_{m+1,m}\boldsymbol{v}_{m+1}\boldsymbol{e}_m^{\mathsf{T}}. \tag{2.3}$$

From $H_m = V_m^{\mathsf{H}} A V_m$, we see that for Hermitian $A = A^{\mathsf{H}}$, the Hessenberg matrix $H_m$ is Hermitian, too, and thus tridiagonal. But if $h_{i,j} = 0$, the vector $\boldsymbol{v}_i$ is already orthogonal to $\boldsymbol{w}_j$ in Algorithm 1. If we simplify the Arnoldi process according to this, we arrive at the *Lanczos process*. We describe it in Algorithm 2, where we overload the function Arnoldi from Algorithm 1. Note that the Arnoldi process has an increasing number of inner products $\boldsymbol{v}_i^{\mathsf{H}}\boldsymbol{w}_j$ in each iteration and so becomes more expensive as the iteration proceeds. In contrast, there is only one inner product (for $i = j$) in the Lanczos process, which leads to a constant amount of computational work per iteration.

**Algorithm 1** Arnoldi process [78, Algorithm 6.2]

**function** ARNOLDI($A$, $\boldsymbol{b}$, $m$)
  $\boldsymbol{v}_1 = \frac{1}{\|\boldsymbol{b}\|}\boldsymbol{b}$
  **for** $j = 1, \ldots, m$ **do**
    $\boldsymbol{w}_j = A\boldsymbol{v}_j$
    **for** $i = 1, \ldots, j$ **do**
      $h_{i,j} = \boldsymbol{v}_i^{\mathsf{H}}\boldsymbol{w}_j$
      $\boldsymbol{w}_j = \boldsymbol{w}_j - h_{i,j}\boldsymbol{v}_i$
    $h_{j+1,j} = \|\boldsymbol{w}_j\|_2$
    **if** $h_{j+1,j} = 0$ **then**
      **break**
    $\boldsymbol{v}_{j+1} = \boldsymbol{w}_j / h_{j+1,j}$
  $V_m = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m]$, $H_m = [h_{i,j}]_{i,j=1,\ldots,m}$
  **return** $V_m$, $H_m$, $h_{m+1,m}$, $\boldsymbol{v}_{m+1}$

**Algorithm 2** Lanczos process [78, Algorithm 6.15]

**function** ARNOLDI($A = A^{\mathsf{H}}$, $\boldsymbol{b}$, $m$)
  $v_1 = \frac{1}{\|\boldsymbol{b}\|}\boldsymbol{b}$
  **for** $j = 1, \ldots, m$ **do**
    **if** $j \geq 2$ **then**
      $\boldsymbol{w}_j = A\boldsymbol{v}_j - h_{j,j-1}\boldsymbol{v}_{j-1}$
    **else**
      $\boldsymbol{w}_j = A\boldsymbol{v}_j$
    $h_{j,j} = \boldsymbol{v}_j^{\mathsf{H}}\boldsymbol{w}_j$
    $\boldsymbol{w}_j = \boldsymbol{w}_j - h_{j,j}\boldsymbol{v}_j$
    $h_{j+1,j} = h_{j,j+1} = \|\boldsymbol{w}_j\|_2$
    **if** $h_{j+1,j} = 0$ **then**
      **break**
    $\boldsymbol{v}_{j+1} = \boldsymbol{w}_j / h_{j+1,j}$
  $V_m = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m]$, $H_m = [h_{i,j}]_{i,j=1,\ldots,m}$
  **return** $V_m$, $H_m$, $h_{m+1,m}$, $\boldsymbol{v}_{m+1}$

**Remark 2.60.** Hermitian matrices are not the only matrices that yield short recurrences in the Arnoldi method, i.e., where only a limited number of diagonals of the Hessenberg matrix $H_m$ is non-zero. See [34] for a full characterization.

We obtain an approximation to $f(A)\boldsymbol{b}$ from the Krylov subspace by evaluating the function at the Hessenberg matrix:

**Definition 2.61.** Let $f$ be defined on the spectrum of $A$ and on the spectrum of $H_m$. We call the approximation

$$f(A)\boldsymbol{b} \approx \boldsymbol{f}_m \coloneqq V_m f(V_m^{\mathsf{H}} A V_m) V_m^{\mathsf{H}} \boldsymbol{b} = \|\boldsymbol{b}\|_2 V_m f(H_m) \boldsymbol{e}_1. \tag{2.4}$$

the *Arnoldi approximation* to $f(A)\boldsymbol{b}$.

As justification for this approximation, consider the following theorem:

**Theorem 2.62** (e.g., [30, Theorem 2.4])**.** *Let $V_m$, $H_m$, $h_{m+1,m}$ and $\boldsymbol{v}_{m+1}$ result from the Arnoldi process for $A$ and $\boldsymbol{b}$. Then*

$$\|\boldsymbol{b}\|_2 V_m f(H_m) \boldsymbol{e}_1 = \|\boldsymbol{b}\|_2 V_m q(H_m) \boldsymbol{e}_1 = q(A)\boldsymbol{b},$$

*where $q$ is the Hermite interpolating polynomial that interpolates $f$ at the spectrum of $H_m$.*

Theorem 2.62 tells us that using the Arnoldi approximation means we use the spectrum of $H_m$ instead of the spectrum of $A$ to construct the interpolating polynomial. Therefore, the approximation is close to the correct solution if the $m$ eigenvalues of the Hessenberg matrix (the so-called *Ritz values*) are close to the $n$ eigenvalues of $A$. From $H_m = V_m^{\mathsf{H}} A V_m$, it follows that the eigenvalues lie in the numerical range of $A$, i.e.,

$$\operatorname{spec}(H_m) \subseteq \mathcal{W}(A) \coloneqq \{\boldsymbol{x}^{\mathsf{H}} A \boldsymbol{x} : \|\boldsymbol{x}\|_2 = 1\}.$$

As we also have $\operatorname{spec}(A) \subseteq \mathcal{W}(A)$, the Arnoldi approximation Eq. (2.4) is a reasonable approach. Furthermore, we know that the eigenvalues of $H_m$ eventually become eigenvalues of $A$. In other words, the Arnoldi approximation is exact after a finite number of iterations:

**Lemma 2.63** ([78, Proposition 6.6], [56, Section 13.2.2])**.** *There exists a smallest number $m^\star \leq n$ such that $\mathcal{K}_{m^\star+1}(A, \boldsymbol{b}) = \mathcal{K}_{m^\star}(A, \boldsymbol{b})$. This is the first $j$ for which $h_{j+1,j} = 0$, i.e., the Arnoldi process is feasible up to $m^\star$ and only then breaks down.*

*We also have*

$$\operatorname{spec}(H_{m^\star}) \subseteq \operatorname{spec}(A), \quad f(A)\boldsymbol{b} = \|\boldsymbol{b}\|_2 V_{m^\star} f(H_{m^\star}) \boldsymbol{e}_1,$$

*i.e., the Arnoldi approximation is exact for $m^\star$.*

Typically, we can only perform a small number of iterations $m_{\max} \ll m^\star$ (see below), so we are more interested in how fast the Ritz values converge to eigenvalues of $A$. We do not discuss this here and refer instead to [79, Chapter 6] and [63].

**Remark 2.64.** When discussing Krylov subspace methods, one often assumes exact arithmetic. This can be problematic especially for the Lanczos process: In finite-precision arithmetic, the assumed orthogonality of the basis vectors can be gradually lost. In this case, $H_m$ is not the orthogonal projection of $A$ onto $\operatorname{span}(\boldsymbol{v}_1, \dots, \boldsymbol{v}_m)$. Round-off errors can thus influence the convergence of the Ritz values to the eigenvalues of $A$. See [70, Section 4] for an overview of theoretical results.

The Arnoldi approximation has the advantage that it needs $A$ only for the matrix-vector products $A\boldsymbol{v}_j$. Thus, it can be applied to cases where $A$ is not explicitly stored. It is of particular use for sparse matrices, where the cost of a matrix-vector product is only $\mathcal{O}(n)$.

However, we need to store the matrix $V_m$ which is generally a full matrix. This is true even when $A$ is sparse or Hermitian (or both).[6] For large $n$, we might be able to store a sparse matrix $A$, but we are able to store $V_m$ only for a small number of iterations $m$ due to limited memory. Moreover, in the non-Hermitian case, the iterations become more expensive with every iteration, which constitutes a computational hazard. There are several approaches to remedy this:

- In the Hermitian case, one can use the *two-pass Lanczos process*. The idea is to construct the Krylov subspace twice. First, we compute the Hessenberg matrix $H_m$ without storing $V_m$. Then, when the size $m$ of the subspace is deemed sufficiently large for a good approximation, we evaluate $\boldsymbol{y} = f(H_m)\boldsymbol{e}_1$. We compute

$$\boldsymbol{f}_m = V_m \boldsymbol{y} = \sum_{j=1}^{m} y_j \boldsymbol{v}_j,$$

  where $y_j$ is the $j$th element of $\boldsymbol{y}$, by constructing the subspace again; this time to obtain the contributions of each basis vector $\boldsymbol{v}_j$ to the approximation $\boldsymbol{f}_m$. Doing so avoids storing $V_m$ but doubles the number of matrix-vector products, which are usually the dominant computational cost.

  A problem is finding a reliable criterion for when to stop increasing $m$ since the approximation $\boldsymbol{f}_m$ is not available after every iteration. In addition, the cost of evaluating $f(H_m)$ still increases with $m$. If a large number $m$ of iterations are needed, then the loss of orthogonality (Remark 2.64) can also slow down convergence.

- We can replace the Krylov subspace by some other subspace so that a small number of iterations is sufficient for a good approximation. In particular, we want to mention rational Krylov subspaces [48]

$$q_{m-1}(A)^{-1}\mathcal{K}_m(A, \boldsymbol{b}),$$

[6] We discuss an exception in Section 2.5.1.

where $q_{m-1}$ is a polynomial of degree $m-1$, and—as a special case—the extended Krylov subspace [61]

$$A^{-m}\mathcal{K}_{2m}(A, \boldsymbol{b}) = \mathcal{K}_m(A, \boldsymbol{b}) + \mathcal{K}_m(A^{-1}, A^{-1}\boldsymbol{b}).$$

Instead of polynomial approximations to $f(A)\boldsymbol{b}$, one now obtains rational approximations, which promise higher accuracy. The disadvantage is that one needs to solve shifted linear systems $(A + sI)^{-1}\boldsymbol{b}$, which might necessitate a Krylov subspace method on its own, cf. Section 2.5.

- Recently, [50] presented a method, which partially avoids the orthogonalization in the Arnoldi process by using a randomized subspace embedding (see, e.g., [67]). The representation of $f(A)$ via a Cauchy integral (see Definition 2.57) is used to reduce the problem $f(A)\boldsymbol{b}$ to shifted inverses $(A+sI)^{-1}\boldsymbol{b}$. Then, Krylov subspace methods for inverses are accelerated by a *sketch-and-solve* approach similar to [74].
- Another approach is given by *restarting* the Arnoldi process, which was first described in [30], cf. [88]. Here, one approximates the error of the Arnoldi approximation by another Arnoldi approximation and continues like this iteratively. We discuss this in more detail in the following.

### 2.4.2. The restarted Arnoldi method

Suppose that we are only able to run $m_{\max}$ iterations of Arnoldi. As explained above, this might be due to a limited amount of memory or because the orthogonalization becomes too expensive in later iterations in the non-Hermitian case. In both cases, the underlying problem is the number of basis vectors $\boldsymbol{v}_j$ that increases with $m$. The Arnoldi approximation Eq. (2.4) might still be far from the exact solution, i.e., the norm $\|\boldsymbol{\varepsilon}_{m_{\max}}\|_2$ of the error

$$\boldsymbol{\varepsilon}_m \coloneqq f(A)\boldsymbol{b} - \boldsymbol{f}_m$$

is larger than the desired accuracy. The idea behind *restarts* is to express the error $\boldsymbol{\varepsilon}_m$ after $m$ iterations[7] again as a matrix function times a vector,

$$\boldsymbol{\varepsilon}_m = f^{(2)}(A)\boldsymbol{b}^{(2)}.$$

Then we can start a new Arnoldi process to approximate this quantity. If the resulting Arnoldi approximation

$$f^{(2)}(A)\boldsymbol{b}^{(2)} \approx \boldsymbol{f}_m^{(2)} \coloneqq \|\boldsymbol{b}^{(2)}\|_2 V_m^{(2)} f^{(2)}(H_m^{(2)})\boldsymbol{e}_1$$

is close to the exact error $f^{(2)}(A)\boldsymbol{b}^{(2)}$, then $\boldsymbol{f}_m + \boldsymbol{f}_m^{(2)}$ is close to $f(A)\boldsymbol{b}$. Doing this iteratively, we obtain a series of approximations

$$f(A)\boldsymbol{b} \approx \boldsymbol{d}_m^{(k)} = \boldsymbol{d}_m^{(k-1)} + \boldsymbol{f}_m^{(k)}, \quad k = 2, 3, \ldots,$$

[7] Restarts are not restricted to $m_{\max}$ and can be applied for smaller $m$. Because of this and for notational simplicity, we write $m$ instead of $m_{\max}$.

where $\boldsymbol{d}_m^{(1)} = \boldsymbol{f}_m^{(1)} = \boldsymbol{f}_m$ is just the original Arnoldi approximation to $f(A)\boldsymbol{b}$ and $\boldsymbol{f}_m^{(k)}$ for $k \geq 2$ is the Arnoldi approximation to

$$f^{(k)}(A)\boldsymbol{b}^{(k)} = \boldsymbol{\varepsilon}_m^{(k-1)} \coloneqq f(A)\boldsymbol{b} - \boldsymbol{d}_m^{(k-1)}.$$

For simplicity, we assume here and in the following that the same number of iterations $m$ is chosen for all Arnoldi decompositions and call $m$ the *restart length*.

Of course, we need representations for $f^{(k)}$ and $\boldsymbol{b}^{(k)}$ as without them we cannot implement this method. We state the resulting algorithm (called *restarted Arnoldi method*) in Algorithm 3 under the assumption that $\boldsymbol{b}^{(k)} = \|\boldsymbol{b}\|_2 \boldsymbol{v}_{m+1}^{(k-1)}$. This is the case in all known approaches, see, e.g., [L, 1, 29, 30, 40, 59].

**Algorithm 3** Generic restarted Arnoldi method for $f(A)\boldsymbol{b}$ (see [L, 30, 40])

**function** RESTARTED_ARNOLDI($f$, $A$, $\boldsymbol{b}$, $m$)
  $V_m^{(1)}$, $H_m^{(1)}$, $h_{m+1,m}^{(1)}$, $\boldsymbol{v}_{m+1}^{(1)}$ = ARNOLDI($A$, $\boldsymbol{b}$, $m$)
  $\boldsymbol{d}_m^{(1)} = \boldsymbol{f}_m^{(1)} = \|\boldsymbol{b}\|_2 V_m^{(1)} f(H_m^{(1)})\boldsymbol{e}_1$
  **for** $k = 2, 3, \ldots$ until convergence **do**
    Determine the error function $f^{(k)}$ such that $\boldsymbol{\varepsilon}_m^{(k-1)} = \|\boldsymbol{b}\|_2 f^{(k)}(A)\boldsymbol{v}_{m+1}^{(k-1)}$.
    $V_m^{(k)}$, $H_m^{(k)}$, $h_{m+1,m}^{(k)}$, $\boldsymbol{v}_{m+1}^{(k)}$ = ARNOLDI($A$, $\boldsymbol{v}_{m+1}^{(k-1)}$, $m$)
    $\boldsymbol{f}_m^{(k)} = \|\boldsymbol{b}\|_2 V_m^{(k)} f^{(k)}(H_m^{(k)})\boldsymbol{e}_1$
    $\boldsymbol{d}_m^{(k)} = \boldsymbol{d}_m^{(k-1)} + \boldsymbol{f}_m^{(k)}$
  **return** $\boldsymbol{d}_m^{(k)}$

The approximation $\boldsymbol{d}_m^{(k)}$ from the restarted Arnoldi method is still a polynomial approximation:

**Lemma 2.65** (see [30])**.** *Let* $W_{km} = [V_m^{(1)} \ldots V_m^{(k)}]$. *Let* $\Upsilon_{km}$ *be defined by the recursion*

$$\Upsilon_{km} = \begin{bmatrix} \Upsilon_{(k-1)m} & \\ h_{m+1,m}^{(k-1)} \boldsymbol{e}_1 \boldsymbol{e}_{(k-1)m}^{\mathsf{T}} & H_m^{(k)} \end{bmatrix}, \quad \Upsilon_m = H_m^{(1)}.$$

*Then*

$$\boldsymbol{d}_m^{(k)} = \|\boldsymbol{b}\|_2 W_{km} f(\Upsilon_{km})\boldsymbol{e}_1 = \|\boldsymbol{b}\|_2 W_{km} q(\Upsilon_{km})\boldsymbol{e}_1 = q(A)\boldsymbol{b},$$

*where* $q$ *is the Hermite interpolating polynomial that interpolates* $f$ *at the spectrum of* $\Upsilon_{km}$.

*Proof.* Combine Eq. (3.14) with Theorem 2.4 in [30]. □

Considering this characterization, it is not surprising that we can always represent $f^{(k)}$ via divided differences:

**Theorem 2.66** ([30, Theorem 2.6])**.** *Define*

$$\gamma_m^{(k)} \coloneqq \prod_{j=1}^{m} h_{j+1,j}^{(k)}.$$

*Let $\theta_1^{(k)}, \dots, \theta_m^{(k)}$ be the eigenvalues of $H_m^{(k)}$ including algebraic multiplicities. Then the error of the restarted Arnoldi approximation to $f(A)\boldsymbol{b}$ after $k \geq 1$ cycles is*

$$\boldsymbol{\varepsilon}_m^{(k)} = f^{(k+1)}(A)\boldsymbol{b}^{(k+1)}$$

*with $\boldsymbol{b}^{(k+1)} = \|\boldsymbol{b}\|_2 \boldsymbol{v}_{m+1}^{(k)}$ and*

$$f^{(k+1)}(s) = \gamma_m^{(k)} \Delta_{[\theta_1^{(k)},\dots,\theta_m^{(k)},s]}\{f^{(k)}\}.$$

It is well known that working with divided differences is prone to numerical instabilities (see, e.g., [69]). It has been observed that this instability can prevent the convergence of the restarted Arnoldi method. Consequently, other error representations (or slightly modified restart algorithms) have been developed in [1, 30, 59]. Yet, none of the known approaches fulfills all of the following properties:

- Numerically stable.
- Generally applicable (i.e., for all kinds of functions and all kinds of matrices).
- Efficient (i.e., the cost of evaluating one cycle in Algorithm 3 is dominated by the Arnoldi decomposition and consequently bounded for $k \to \infty$).

An error representation that comes close to fulfilling all three requirements was presented in [40]. There, a representation based on an integral representation of $f$ was developed.

**Theorem 2.67** ([40, Corollary 3.5])**.** *Let the analytic function $f : \Omega \to \mathbb{C}$ have the integral representation*

$$f(s) = \int_\Gamma \frac{g(t)}{t-s}\,\mathrm{d}t$$

*with a path $\Gamma \subseteq \mathbb{C}\backslash\Omega$ and a function $g : \Gamma \to \mathbb{C}$. Let $w_m^{(j)}(t) = \det(tI - H_m^{(j)})$ and $\gamma_m^{(j)}$ be as in Theorem 2.66. If further $\mathcal{W}(A) \subseteq \Omega$, then*

$$f^{(k+1)}(s) = \gamma_m^{(1)} \dots \gamma_m^{(k)} \int_\Gamma \frac{g(t)}{w_m^{(1)}(t) \dots w_m^{(k)}(t)} (t-s)^{-1}\,\mathrm{d}t,$$

*provided the integrals exist.*

After choosing a suitable quadrature rule, this error representation allows for stable and efficient restarts for many practically relevant functions $f$. In particular, Theorem 2.67 includes the case where $\Gamma$ is a closed contour with $\mathcal{W}(A)$ in its interior. Then $g(t) = f(t)(2\pi i)^{-1}$, cf. Definition 2.57. When applying Theorem 2.67, choosing $\Gamma$ and the quadrature rule typically requires detailed knowledge about $\mathcal{W}(A)$. For Stieltjes functions, however, the situation simplifies.

**Theorem 2.68** ([L, Theorem 2.1], cf. [39, Theorem 2.1, Eq. (2.4)])**.** *Let $f$ be a Stieltjes function (Definition 2.26), i.e.,*

$$f(s) = \int_0^\infty \frac{\rho(t)}{t+s}\,\mathrm{d}t.$$

*Assume that $\mathcal{W}(A) \cap (-\infty, 0] = \varnothing$. Let $\psi_m^{(j)}(t) = \boldsymbol{e}_m^\mathsf{T}(H_m^{(j)} + tI)^{-1}\boldsymbol{e}_1$. Then*

$$f^{(k+1)}(s) = (-1)^k\Big(\prod_{j=1}^k h_{m+1,m}^{(j)}\Big)\int_0^\infty \rho(t)\Big(\prod_{j=1}^k \psi_m^{(j)}(t)\Big)(s+t)^{-1}\,\mathrm{d}t.$$

*Proof.* While the theorem was already stated in [L, Theorem 2.1] and slightly modified in [39, Theorem 2.1, Eq. (2.4)], no full proof was given in either case. Furthermore, the error representations in [L] and [39] differ by a factor of $(-1)^k$. We include the proof here to show that the one in [L] is correct.

We use the fact that Theorem 2.67 includes Stieltjes functions in the following way: We start from the integral representation of $f$ and use the transformation $\tau = -t$. Then we see that $f$ has the integral representation assumed in Theorem 2.67,

$$f(s) = \int_0^\infty \frac{\rho(t)}{t+s}\,\mathrm{d}t = \int_{-\infty}^0 \frac{\rho(-\tau)}{s-\tau}\,\mathrm{d}\tau = \int_{-\infty}^0 \frac{-\rho(-\tau)}{\tau-s}\,\mathrm{d}\tau,$$

with $\Gamma = (-\infty, 0]$ and $g(t) = -\rho(-t)$. Thus, it holds by Theorem 2.67

$$\begin{aligned}
f^{(k+1)}(s) &= -\gamma_m^{(1)}\dots\gamma_m^{(k)}\int_{-\infty}^0 \frac{\rho(-t)}{w_m^{(1)}(t)\dots w_m^{(k)}(t)}(t-s)^{-1}\,\mathrm{d}t \\
&= -\gamma_m^{(1)}\dots\gamma_m^{(k)}\int_0^\infty \frac{\rho(\tau)}{w_m^{(1)}(-\tau)\dots w_m^{(k)}(-\tau)}(-\tau-s)^{-1}\,\mathrm{d}\tau \\
&= \gamma_m^{(1)}\dots\gamma_m^{(k)}\int_0^\infty \frac{\rho(t)}{w_m^{(1)}(-t)\dots w_m^{(k)}(-t)}(t+s)^{-1}\,\mathrm{d}t
\end{aligned}$$

by using again the transformation $\tau = -t$ for the second line and renaming $\tau = t$ in the third line. Next, note that

$$\begin{aligned}
\frac{\gamma_m^{(j)}}{w_m^{(j)}(-t)} &= h_{m+1,m}^{(j)}\boldsymbol{e}_m^\mathsf{T}(-tI - H_m^{(j)})^{-1}\boldsymbol{e}_1 = -h_{m+1,m}^{(j)}\boldsymbol{e}_m^\mathsf{T}(tI + H_m^{(j)})^{-1}\boldsymbol{e}_1 \\
&= -h_{m+1,m}^{(j)}\psi_m^{(j)}(t).
\end{aligned}$$

The first equality is easily verified by Cramer's rule and was already stated, e.g., in [40, Eq. (3.20)]. Applying this in the integral above yields the theorem's statement. □

**Remark 2.69.** We assume that the number of iterations $m$ in each restart cycle is a constant. It is, however, possible to assign each cycle $j$ an individual restart length $m^{(j)}$. Then, one might wonder how to choose $k$ and $m^{(1)}, \dots, m^{(k)}$ such that the number of matrix-vector products $\sum_{j=1}^{k} m^{(j)}$ is minimized while $\|\boldsymbol{\varepsilon}^{(k)}\|_2$ is below the desired tolerance. One might expect that choosing $m^{(j)} = m_{\max}$ as large as possible yields the best result. However, it was observed in the context of restarted Krylov subspace methods that increasing $m$ can hurt [31]. Numerical experiments by Marcel Schweitzer (personal communication, University of Wuppertal, Oct 2022) show that for the algorithm in [40] (i.e., using Theorems 2.67 and 2.68), it can in some cases be beneficial to choose

$$m^{(j)} = \begin{cases} m_{\max} & \text{if } j \text{ odd}, \\ \hat{m} & \text{else} \end{cases}$$

with $m_{\max} > \hat{m}$ instead of $\hat{m} = m_{\max}$. This topic is probably related to the Forsythe conjecture [33, 36] but has not received much attention in the literature and goes beyond the scope of this thesis.

### 2.4.3. Error bounds for the restarted Arnoldi method

Lemma 2.65 reveals that the restarted Arnoldi method approximates $f(A)\boldsymbol{b}$ by interpolating $f$ at the eigenvalues of $\Upsilon_{km}$, i.e., the eigenvalues of $\operatorname{diag}(H_m^{(1)}, \dots, H_m^{(k)})$. We know from Lemma 2.63 that the eigenvalues of $H_m^{(k)}$ are eigenvalues of $A$ if $m$ is large enough, in which case the approximation is exact. If $m$ is not large enough and we increase $k$ instead, the eigenvalues of $\Upsilon_{km}$ do not converge to the eigenvalues of $A$, however. Thus, even for $km \geq n$, we do not obtain the exact solution from the restarted Arnoldi method. It can still be attractive if we get numerically close to the exact solution after a reasonable number $km$ of matrix-vector products. Thus, an important question is whether we can obtain convergence results.

#### A priori bounds

We start with *a priori* error bounds, i.e., bounds that are useful before starting a calculation. They generally show that under certain conditions

$$\lim_{k\to\infty} \boldsymbol{d}_m^{(k)} = f(A)\boldsymbol{b}.$$

While *a priori* bounds also bound the rate of convergence, they can be very pessimistic in this regard. We give two error bounds relevant for our investigation.

As Lemma 2.65 shows us that we effectively interpolate $f$, one might be inclined to apply results from interpolation theory. The first result is for entire functions of order one (see Definition 2.18).

**Theorem 2.70** ([30, Theorem 4.2])**.** *Let $f$ be an entire function of order one. Let $\boldsymbol{\varepsilon}_m^{(k)}$ denote the error of restarted Arnoldi after $k$ restarts with restart length $m$. Then there exist constants $C$ and $\gamma$ such that the error satisfies*

$$\|\boldsymbol{\varepsilon}_m^{(k)}\|_2 = \|f(A)\boldsymbol{b} - \boldsymbol{d}_m^{(k)}\|_2 \leq C\frac{\gamma^{km-1}}{(km-1)!}.$$

Essentially, this shows that we have guaranteed superlinear convergence to the exact solution for entire functions[8]. The term $\gamma^{km-1}$, however, tells us that this behavior might emerge only after a large number of matrix-vector products $km$ if $\gamma > 1$.

If we restrict ourselves to Stieltjes functions and Hermitian positive definite matrices, we have at least linear convergence.

**Theorem 2.71** ([39, Theorem 4.3])**.** *Let $A$ be Hermitian positive definite and $f$ be a Stieltjes function. Denote by $\kappa(A+tI)$ the condition number of $A+tI$. Define for $t \geq 0$*

$$q(t) = \frac{\sqrt{\kappa(A+tI)}-1}{\sqrt{\kappa(A+tI)}+1}, \quad \gamma_m(t) = \frac{1}{\cosh(m\log q(t))} = \frac{2}{q(t)^{-m}+q(t)^m} < 1.$$

*The error of the restarted Arnoldi method then fulfills*

$$\|\boldsymbol{\varepsilon}_m^{(k)}\|_2 \leq C\gamma_m(t_0)^k.$$

*Here $t_0 \geq 0$ is the left endpoint of the support of $\rho(t)$,*

$$C = \|\boldsymbol{b}\|_2\sqrt{\kappa(A)}\; f(\sqrt{\lambda_{\min}\lambda_{\max}})$$

*and $\lambda_{\min}$ and $\lambda_{\max}$ are the smallest and the largest eigenvalue of $A$, respectively.*

### A posteriori bounds

*A posteriori* bounds are computed while the algorithm is executed. As they are commonly close to the exact error norm, they can be used as a stopping criterion. In [41, 81], one such bound was presented for Stieltjes functions. Reusing an intermediate result of theirs, we later show that it holds for many Laplace transforms, too.

Assume we use the restarted Arnoldi method for $f(A)\boldsymbol{b}$ for some function $f$ and Hermitian positive definite matrix $A$. If we express the error as the action of a matrix function, then

$$\begin{aligned}\|\boldsymbol{\varepsilon}_m^{(k)}\|_2^2 &= (\boldsymbol{\varepsilon}_m^{(k)})^{\mathsf{H}}\boldsymbol{\varepsilon}_m^{(k)} \\ &= (\boldsymbol{v}_{m+1}^{(k)})^{\mathsf{H}} f^{(k+1)}(A)^{\mathsf{H}} f^{(k+1)}(A)\boldsymbol{v}_{m+1}^{(k)} \\ &= (\boldsymbol{v}_{m+1}^{(k)})^{\mathsf{H}}(f^{(k+1)}(A))^2\boldsymbol{v}_{m+1}^{(k)}.\end{aligned}$$

[8] It is mentioned in [30] that the error bound can be generalized to entire functions of other orders.

Functions of the form $\boldsymbol{v}_1^{\mathsf{H}} g(A)\boldsymbol{v}_2$ for some vectors $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ and a matrix function $g(A)$ are known as *sesquilinear forms*, which correspond to *bilinear forms* in the real case. In [46], it is explained how to compute bilinear forms or bounds for them using the Lanczos process. Based on this, one can derive the following result:

**Theorem 2.72** (cf. [41, Theorem 4])**.** *Let $A$ be Hermitian positive definite and $g$ be a completely monotone function. Denote by $H_m$ the tridiagonal matrix obtained from the Lanczos process applied to $A$ and $\boldsymbol{v}$ after $m$ steps. Let*

$$\widetilde{H}_m = \begin{bmatrix} H_m & h_{m+1,m}\boldsymbol{e}_m \\ h_{m+1,m}\boldsymbol{e}_m^{\mathsf{H}} & h_{m+1,m}^2 \boldsymbol{e}_m^{\mathsf{T}}(H_m - aI)^{-1}\boldsymbol{e}_m \end{bmatrix}$$

*with $0 < a \leq \min \operatorname{spec}(A)$. Then*

$$\boldsymbol{e}_1^{\mathsf{T}} g(H_m)\boldsymbol{e}_1 \leq \boldsymbol{v}^{\mathsf{H}} g(A)\boldsymbol{v} \leq \boldsymbol{e}_1^{\mathsf{T}} g(\widetilde{H}_m)\boldsymbol{e}_1.$$

*Proof.* The statement essentially coincides with Theorem 4 in [41] if $g$ is a Stieltjes function. However, their derivation only uses that Stieltjes functions are completely monotone and it contains our statement as an intermediate result. In particular, combine Eq. (11), Theorem 1 and Theorem 2 in [41]. □

The bounds of [41] follow by noting that if $f$ is a Stieltjes function, then $f^{(k)}$ is again a Stieltjes function by Theorem 2.68. Moreover, we know that Stieltjes functions are completely monotone by Corollary 2.28 and that the product of two completely monotone functions is again completely monotone by Lemma 2.25.

**Corollary 2.73** ([41, Theorem 4])**.** *Let $A$ be Hermitian positive definite and $f$ be a Stieltjes function. Denote by $H_m^{(k)}$ the tridiagonal matrix from the restarted Arnoldi method for $f(A)\boldsymbol{b}$ after $k$ cycles with restart length $m$ and by $\boldsymbol{\varepsilon}_m^{(k-1)} = f^{(k)}(A)\boldsymbol{v}_m^{(k)}$ the error after $k-1$ cycles. Let $\widetilde{H}_m^{(k)}$ be the modification of $H_m^{(k)}$ as in Theorem 2.72. Then*

$$\|f^{(k)}(H_m^{(k)})\boldsymbol{e}_1\|_2 \leq \|\boldsymbol{\varepsilon}_m^{(k-1)}\|_2 \leq \|f^{(k)}(\widetilde{H}_m^{(k)})\boldsymbol{e}_1\|_2.$$

In practice, we need to compute the Hessenberg matrix from the Lanczos process for $A$ and $\boldsymbol{v}_{m+1}^{(k+1)}$. This does not incur any additional cost, however, for that is exactly what we do if we want to continue the restarted Arnoldi method for an additional restart cycle. We also need a suitable value for $a$. If the smallest eigenvalue of $A$ is not known, one can use an estimate, e.g., the smallest Ritz value, and multiply it by a safety factor $0 < c < 1$. In essence, we can bound the error of the *previous* cycle basically for free.

## 2.5. Iterative methods for rational matrix functions

In this section, we summarize some methods used for computing $f(A)\boldsymbol{b}$ when $f = r$ is a rational function. In Chapter 3, we compare our new method to such methods. Theorem 2.75 is also used in Section 4.4.

We restrict ourselves to the partial fraction expansion (Theorem 2.13): For any rational function $r$, we find

$$r(A)\boldsymbol{b} = p_0(A)\boldsymbol{b} + \sum_{j=1}^{k} \sum_{i=1}^{m_j} w_{j,i}(A - \tau_j I)^{-i}\boldsymbol{b} \tag{2.5}$$

for some $p_0$, $k$, $m_j$ and $w_{j,i}$ according to Theorem 2.13. For evaluating $r(A)\boldsymbol{b}$, we now need to evaluate $p_0(A)\boldsymbol{b}$, which is trivial, and terms of the form $(A - \tau_j I)^{-i}\boldsymbol{b}$. These can be written as

$$(A - \tau_j I)^{-i}\boldsymbol{b} = \underbrace{(A - \tau_j I)^{-1} \dots (A - \tau_j I)^{-1}}_{i \text{ times}} \boldsymbol{b}.$$

It is thus sufficient to discuss how linear systems

$$\boldsymbol{x} = M^{-1}\boldsymbol{b} \iff M\boldsymbol{x} = \boldsymbol{b}$$

can be solved.

Before we start, however, a note on the partial fraction expansion: While methods for linear systems can be used for rational functions via the partial fraction expansion, that does not mean that this is necessarily the best way. If the rational function is given in the form of two polynomials $r(s) = \frac{p(s)}{q(s)}$, we could in principle evaluate $r(A)\boldsymbol{b}$ as

$$r(A)\boldsymbol{b} = p(A)(q(A)^{-1}\boldsymbol{b}) = q(A)^{-1}(p(A)\boldsymbol{b}),$$

which leads to only one linear system we need to solve. In some cases, this is a suitable approach. The partial fraction expansion offers some advantages, however: First, for a large matrix $A$ and a large degree of $q$, it is prohibitively expensive to explicitly form $q(A)$. For some methods like Krylov subspace methods, we only need it implicitly, i.e., its matrix-vector products $q(A)\boldsymbol{v}$, but these products can be much more expensive than, say, the product $(A-\tau_j I)\boldsymbol{v}$. Second, the condition number of $q(A)$ might be much larger than the one of $A$ or $(A - \tau_j I)$. Third, the partial fraction expansion approach is *embarrassingly parallelizable*: The sums

$$\sum_{i=1}^{m_j} w_{j,i}(A - \tau_j I)^{-i}\boldsymbol{b}$$

for $j = 1, \dots, k$ can be evaluated separately from each other. Moreover, we explain in Section 2.5.1 that Krylov subspace methods can solve these systems very efficiently.

### 2.5.1. Krylov subspace methods for the matrix inverse

We already discussed in Section 2.4.1 how the Krylov subspace $\mathcal{K}_m(A, \boldsymbol{b})$ can be used for an approximation to $f(A)\boldsymbol{b}$. We now consider $f(s) = s^{-1}$. Then, the Arnoldi approximation to the problem

$$\boldsymbol{x} = A^{-1}\boldsymbol{b} \iff A\boldsymbol{x} = \boldsymbol{b}$$

is given by

$$\boldsymbol{x}_m = \|\boldsymbol{b}\|_2 V_m H_m^{-1} \boldsymbol{e}_1.$$

This is known as the *full orthogonalization method (FOM)*. Note that it is often customary to provide an initial guess $\boldsymbol{x}_0$ if $f(s) = s^{-1}$ and to construct the Krylov subspace with the residual $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$ instead of $\boldsymbol{b}$. We assume in the following that $\boldsymbol{x}_0 = 0$. Thus $\boldsymbol{r}_0 = \boldsymbol{b}$.

FOM as we have described it here is rarely used in practice. It suffers from the same problem as the Arnoldi approximation in general: the arithmetic and memory cost both increase with $m$. Restarts are possible since as a simplification compared to the general case, the error function can be formulated immediately using the residual:

**Lemma 2.74** ([78, Proposition 6.7])**.** *Define $\psi_m = \boldsymbol{e}_m^\mathsf{T} H_m^{-1} \boldsymbol{e}_1$ for $m \geq 1$. The residual $\boldsymbol{r}_m = \boldsymbol{b} - A\boldsymbol{x}_m$ of FOM fulfills*

$$\boldsymbol{r}_m = -h_{m+1,m} \|\boldsymbol{b}\|_2 \psi_m \boldsymbol{v}_{m+1}.$$

Now we can easily see that

$$\boldsymbol{\varepsilon}_m = A^{-1}\boldsymbol{b} - \boldsymbol{x}_m = A^{-1}(\boldsymbol{b} - A\boldsymbol{x}_m) = A^{-1}\boldsymbol{r}_m = -h_{m+1,m}\|\boldsymbol{b}\|_2 \psi_m A^{-1}\boldsymbol{v}_{m+1},$$

so we have

$$f^{(k)}(s) = (-1)^k \Big( \prod_{i=1}^{k} h_{m+1,m}^{(i)} \psi_m^{(i)} \Big) s^{-1}$$

in Section 2.4.2 with $\psi_m^{(i)} = \boldsymbol{e}_m^\mathsf{T} (H_m^{(i)})^{-1} \boldsymbol{e}_1$ (cf. Theorem 2.68). However, one of the two following variants of FOM is usually used instead.

### Conjugate Gradient

The *Conjugate Gradient method (CG)* is a variant of FOM for Hermitian positive definite matrices. As the matrix is Hermitian, we can use the Lanczos process (Algorithm 2) instead of the Arnoldi process (Algorithm 1) to construct the Krylov subspace. The algorithm for CG is optimized by exploiting this short recurrence and the fact that $f^{(k)}$ is only a scaled version of $f$: The approximation $\boldsymbol{x}_m$ is obtained as a cheap update $\boldsymbol{x}_m = \boldsymbol{x}_{m-1} + \boldsymbol{y}_m$, where $\boldsymbol{y}_m$ (and thus $\boldsymbol{x}_m$) is obtained without explicitly inverting $H_m$. In fact, $H_m$ and $V_m$ are not explicitly formed or stored. This means the amount of work per iteration is constant and only a small amount of memory is needed. See, e.g., [78, Section 6.7] for more details. Note that the only changes are algorithmic. Mathematically, CG is still equivalent to FOM (for Hermitian positive definite matrices).

One can show that with CG the error in the $A$-norm is minimized (e.g., Lemma 2.79 or [47, Theorem 2.3.2]) and decreases strictly monotonically with each iteration (e.g., [64, Theorem 5.6.1]). Moreover, one can show the following bound that was used for Theorem 2.71:

**Theorem 2.75** ([39, Theorem 3.1], [47, Theorem 3.1.1])**.** *Let $A$ be Hermitian positive definite. Denote by $\lambda_{\max}$ and $\lambda_{\min}$ its largest and smallest eigenvalue, respectively, and by $\kappa(A) = \frac{\lambda_{\max}}{\lambda_{\min}}$ its condition number. Then the error $\boldsymbol{\varepsilon}_m$ after $m$ steps of CG for $A\boldsymbol{x} = \boldsymbol{b}$ fulfills*

$$\|\boldsymbol{\varepsilon}_m\|_A \leq \gamma_m \|\boldsymbol{x}\|_A$$

*with*

$$\gamma_m = \frac{1}{\cosh(m \log q)}, \quad q = \frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1}.$$

Since $\cosh(m \log q) = \frac{1}{2}(q^m + q^{-m})$, we can also write this statement in the more familiar (but less tight) form

$$\frac{\|\boldsymbol{\varepsilon}_m\|_A}{\|\boldsymbol{x}\|_A} \leq 2 \left( \frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^m.$$

One should keep in mind that—like the Arnoldi method—FOM and CG compute the exact solution after a finite number of steps. Error bounds such as Theorem 2.75 can thus be very pessimistic. In particular, while a large condition number $\kappa(A)$ implies a large bound, it does not imply a large error. See, e.g., [65, Section 3.1.1] for more details. Moreover, CG often shows superlinear convergence (e.g., [64, Section 5.6.4]). Considering the low computational and memory cost, too, we see that there is no reason to use restarts for CG. See [64, Section 5.9], [70, Section 5] for overviews of the behavior of CG in the presence of round-off errors.

#### Generalized Minimal Residual

If the matrix $A$ is not Hermitian positive definite, then the $A$-norm does not exist. The *Generalized Minimal Residual method (GMRES)* (see, e.g., [78, Section 6.5]) uses the Krylov subspace to minimize the residual norm $\|\boldsymbol{r}_m\|_2$ instead: If the matrix $A$ is non-singular, then this is equivalent to minimizing the error in the $A^{\mathsf{H}}A$-norm,

$$\|\boldsymbol{r}_m\|_2 = \|A\boldsymbol{\varepsilon}_m\|_2 = \sqrt{\boldsymbol{\varepsilon}_m^{\mathsf{H}} A^{\mathsf{H}} A \boldsymbol{\varepsilon}_m} = \|\boldsymbol{\varepsilon}_m\|_{A^{\mathsf{H}}A}.$$

The method is related to FOM in the following way: In FOM, we solve the linear system $H_m \boldsymbol{y}_m = \|\boldsymbol{b}\|_2 \boldsymbol{e}_1$. If $H_m$ is non-singular, then the unique solution is $\boldsymbol{y}_m = \|\boldsymbol{b}\|_2 H_m^{-1} \boldsymbol{e}_1$. This can also be expressed as solving the least-squares problem $\boldsymbol{y}_m = \arg\min_{\boldsymbol{y}} \|\|\boldsymbol{b}\|_2 \boldsymbol{e}_1 - H_m \boldsymbol{y}\|_2$.[9] In GMRES, we extend $H_m$ to

$$\overline{H}_m \coloneqq \begin{bmatrix} H_m \\ \boldsymbol{e}_m^{\mathsf{T}} h_{m+1,m} \end{bmatrix}$$

and solve the least-squares problem $\arg\min_{\boldsymbol{y}} \|\|\boldsymbol{b}\|_2 \boldsymbol{e}_1 - \overline{H}_m \boldsymbol{y}\|_2$ instead. We refer to [78, Section 6.5.3] for some implementational aspects. As the matrices $H_m$ and $\overline{H}_m$ differ in

[9] If $H_m$ is non-singular, the unique minimizer $\boldsymbol{y}_m$ yields $\|\|\boldsymbol{b}\|_2 \boldsymbol{e}_1 - H_m \boldsymbol{y}_m\|_2 = 0$.

essence only by the entry $h_{m+1,m}$, the residual norms produced by FOM and GMRES are closely connected. See, e.g., [78, Section 6.5.7] for more information.

Since it relies on the Arnoldi process, the computational burden of GMRES increases with each iteration. While it deviates slightly from the Arnoldi approximation in Section 2.4.1, it is still possible to use restarts: Given the approximation $\boldsymbol{x}_m$ from GMRES, one would try to solve the system $A(\boldsymbol{x} - \boldsymbol{x}_m) = \boldsymbol{r}_m$ in the next cycle.

#### Multi-shift variants

The partial fraction expansion offers another advantage for Krylov subspace methods. Let us assume for illustration that $m_j = 1$ for all $j$ in Eq. (2.5). Then we only need to solve

$$(A - \tau_j I)^{-1}\boldsymbol{b}$$

for $k$ values of $\tau_j$. The Krylov subspaces are, however, shift-invariant,

$$\mathcal{K}_m(A - \tau_1 I, \boldsymbol{b}) = \mathcal{K}_m(A - \tau_2 I, \boldsymbol{b}) = \cdots = \mathcal{K}_m(A - \tau_k I, \boldsymbol{b}) = \mathcal{K}_m(A, \boldsymbol{b}).$$

This is easily verified from $(A - \tau_j I)\boldsymbol{b} = A\boldsymbol{b} - \tau_j \boldsymbol{b} \in \mathcal{K}_2(A, \boldsymbol{b})$. Because the Krylov subspaces coincide for all linear systems we are interested in, the computed basis of the subspace of one system can be reused for the other ones.[10] Constructing the solution for all systems is thus only marginally more expensive than constructing the solution for a single system. In particular, we only need one matrix-vector product (instead of $k$) to extend the subspaces. Variants of Krylov subspace methods that exploit this are occasionally called *multi-shift*. See [84, Section 14.1] for an overview.

We see from Lemma 2.74 that the residual from FOM is collinear to the last Arnoldi vector $\boldsymbol{v}_{m+1}$. This does not change by introducing a shift. Thus, the Krylov subspaces for several shifted systems also coincide after restarts. A multi-shift variant of restarted GMRES is (only) possible if one relaxes the minimization properties for the shifted systems, see [38].

### 2.5.2. Algebraic multigrid methods

Multigrid methods are another way of solving linear systems. They can be classified as either *geometric* or *algebraic*. The geometric and original point of view comes from the discretization of a differential equation which defines the linear system. In geometric multigrid methods, this system is solved with the help of smaller systems that arise from coarser discretizations. Here, we focus on *algebraic multigrid*, which does not need a geometric interpretation but uses the same terminology. We do not include many practical details as we are mostly interested in the theoretically best possible way of

[10] The interpretation here is that we evaluate the function $f(s) = s^{-1}$ at several matrices $A - \tau_j I$ and it turns out their subspaces coincide. Another interpretation is that we evaluate several functions $f_j(s) = (s - \tau_j)^{-1}$ at the same matrix $A$. This way it is clear that only one basis is needed but algorithms such as CG and GMRES are traditionally described for $f(s) = s^{-1}$ without any shift.

constructing such a method. As is customary, we assume that $A$ is symmetric positive definite. For more information about multigrid methods, we refer to [90] for geometric multigrid and to [95] for algebraic multigrid as starting points.

For illustration purposes, we describe two-grid methods first: Multigrid methods are easily obtained by recursing on two-grid methods. A two-grid method has two components:

- *Smoother*: This is an iterative method that initially decreases the error of the current approximation at low computational cost. However, using only this method would be too expensive, e.g., because asymptotically it decreases the error only very slowly.
- *Coarse-grid correction*: The coarse-grid correction computes an approximation to the error by restricting the residual to a lower-dimensional space, solving for the error there and prolongating it back to the original space. Typically, one effectively applies an $A$-orthogonal projection to the error. The subspace is chosen such that (hopefully) the smoother coupled with this correction converges fast.

Both components update the iterate $\boldsymbol{x}_k$ according to

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \mathfrak{I}(A)^{-1}\boldsymbol{r}_k, \quad k \geq 0, \tag{2.6}$$

where $\boldsymbol{r}_k = \boldsymbol{b} - A\boldsymbol{x}_k$ is the residual. $\mathfrak{I}(A)^{-1} \in \mathbb{R}^{n\times n}$ is a matrix that approximates $A^{-1}$. For smoothers, $\mathfrak{I}(A)^{-1}$ is typically defined by its inverse $\mathfrak{I}(A) \coloneqq (\mathfrak{I}(A)^{-1})^{-1}$. For a coarse-grid correction, $\mathfrak{I}(A)^{-1}$ is a projector and thus singular. In this case, we consider $\mathfrak{I}(A)^{-1}$ a single symbol.

### Smoother

We assume that a stationary method is used, i.e., we can write the error as $\boldsymbol{\varepsilon}_{k+1} = E^{k+1}\boldsymbol{\varepsilon}_0$ with the *error propagator* $E = I - \mathfrak{I}(A)^{-1}A$. These are typically splitting-based methods like Jacobi or Gauß-Seidel: One defines two matrices $M$, $N$ such that $A = M - N$ and $M$ is non-singular. Then the method is obtained by setting $\mathfrak{I}(A)^{-1} = M^{-1}$ in Eq. (2.6). Note that we can construct any non-singular matrix $M$ in this way. A good splitting-based method tries to set $N$ close to 0 while ensuring that $M$ can be cheaply inverted.

**Example 2.76.** We can easily represent the common splitting-based methods using the element-wise (or Hadamard) product $\odot$:

$$\begin{aligned} \text{Jacobi:} \quad & \mathfrak{I}(A) = A \odot I, \\ \text{Gauß-Seidel:} \quad & \mathfrak{I}(A) = A \odot \begin{bmatrix} 1 & & \\ \vdots & \ddots & \\ 1 & \dots & 1 \end{bmatrix}. \end{aligned}$$

The respective block methods can also be represented in this way. For example, the block Jacobi method with block size 2 would lead to

$$\mathfrak{I}(A) = A \odot \begin{bmatrix} \mathbb{1}_2 & & \\ & \ddots & \\ & & \mathbb{1}_2 \end{bmatrix}, \quad \mathbb{1}_2 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

assuming that $n$ is even.

We see that splitting-based methods are stationary from

$$\begin{aligned} \boldsymbol{\varepsilon}_{k+1} = A^{-1}\boldsymbol{b} - \boldsymbol{x}_{k+1} &= A^{-1}\boldsymbol{b} - \boldsymbol{x}_k - \mathfrak{I}(A)^{-1}\boldsymbol{r}_k \\ &= \boldsymbol{\varepsilon}_k - \mathfrak{I}(A)^{-1}A\boldsymbol{\varepsilon}_k \\ &= E\boldsymbol{\varepsilon}_k = E^{k+1}\boldsymbol{\varepsilon}_0. \end{aligned}$$

Now consider some[11] norm of the error. We have[12]

$$\begin{aligned} \|\boldsymbol{\varepsilon}_k\| = \|E^k\boldsymbol{\varepsilon}_0\| \leq \|E^k\|\|\boldsymbol{\varepsilon}_0\|, \\ \lim_{k\to\infty} \|E^k\|^{1/k} = \rho(E). \end{aligned}$$

Consequently, for large $k$, we expect that a stationary method behaves according to the spectral radius $\rho(E)$ of the error propagator. This leads to the following famous theorem.

**Theorem 2.77** (e.g., [78, Theorem 4.1])**.** *Let $\rho(E) < 1$. Then the splitting-based method converges for any right-hand side $\boldsymbol{b}$ and initial guess $\boldsymbol{x}_0$.*

Many results exist that guarantee $\rho(E) < 1$ for different methods and classes of matrices. See, e.g., [96, Chapter 4].

For small values of $k$, however, we usually see a better reduction of the error norm than $\rho(E)$. The reason is simple: The modulus of many eigenvalues lies far from $\rho(E)$ and closer to 0. Generally, the initial error $\boldsymbol{\varepsilon}_0$ contains contributions from eigenvectors with eigenvalue close to 0. Thus, we observe an initial rate of convergence close to 0. However, with each iteration, these contributions become less significant,[13] which is why the rate of convergence can slow down significantly.

**Remark 2.78.** In geometric multigrid (for elliptic partial differential equations), the eigenvectors of $E$ with eigenvalues close to 0 correspond to high frequencies, i.e., their value changes quickly when going from one variable to the next. Consequently, the problematic eigenvectors and the error after a few iterations look geometrically smooth, hence the name "smoother".

[11] For now, it does not matter which norm is used.

[12] The limit is known as the *Gelfand formula*, see, e.g., [58, Corollary 5.6.14].

[13] For a formal description of this effect, see the *Power Method* (e.g., [79, Section 4.1.1]) for eigenvalue approximation.

### Motivation for a coarse-grid correction

We saw that the rate of convergence of typical stationary methods is often very close to 1 for large values of $k$. Using only such a method would therefore not be a good idea. The problem is that the error $\boldsymbol{\varepsilon}_k$ is composed mostly of a subset of the problematic eigenvectors. An idea for solving this is to orthogonalize the error against the problematic eigenvectors. This would necessitate the error itself but if we knew the error, we would also have solved the linear system. Instead, we apply a projection to the residual $\boldsymbol{r}_k = \boldsymbol{b} - A\boldsymbol{x}_k$, which can be readily calculated.

Let us assume that the matrix $\mathfrak{I}(A) = M$ of the smoother is symmetric, $M = M^{\mathsf{T}}$. As $A$ is symmetric positive definite, we know that the generalized eigendecomposition

$$MQ = AQD$$

yields $Q$ such that $Q^{\mathsf{T}}AQ = QQ^{\mathsf{T}}A = I$ and a real diagonal $D$, see Theorem 2.86. The columns of $Q$ are eigenvectors of the error propagator $E$, i.e.,

$$MQ = AQD \iff QD^{-1} = M^{-1}AQ \iff (I - M^{-1}A)Q = Q\Lambda$$

with $\Lambda = I - D^{-1}$. We can thus rewrite the error in terms of the residual,

$$\boldsymbol{\varepsilon}_{k+1} = (I - M^{-1}A)\boldsymbol{\varepsilon}_k = (I - M^{-1}A)QQ^{\mathsf{T}}A\boldsymbol{\varepsilon}_k = Q\Lambda Q^{\mathsf{T}}\boldsymbol{r}_k.$$

Let us now consider the error in the $A$-norm:

$$\|\boldsymbol{\varepsilon}_{k+1}\|_A^2 = \boldsymbol{\varepsilon}_{k+1}^{\mathsf{T}}A\boldsymbol{\varepsilon}_{k+1} = \boldsymbol{r}_k^{\mathsf{T}}Q\Lambda Q^{\mathsf{T}}AQ\Lambda Q^{\mathsf{T}}\boldsymbol{r}_k = \|\Lambda Q^{\mathsf{T}}\boldsymbol{r}_k\|_2^2.$$

Since

$$\Lambda Q^{\mathsf{T}}\boldsymbol{r}_k = \begin{bmatrix} \lambda_1 \boldsymbol{v}_1^{\mathsf{T}}\boldsymbol{r}_k \\ \vdots \\ \lambda_n \boldsymbol{v}_n^{\mathsf{T}}\boldsymbol{r}_k \end{bmatrix},$$

an eigenvalue $\lambda_i$ of $E$ contributes to the $A$-norm of the error only if $\boldsymbol{v}_i^{\mathsf{T}}\boldsymbol{r}_k \neq 0$. This motivates the orthogonalization of the residual $\boldsymbol{r}_k$ against problematic eigenvectors before applying the smoother. We achieve this by the *Galerkin* approach. We state it here more generally for the complex case.

**Lemma 2.79** ([90, Corollary A.2.1])**.** *Let $\boldsymbol{x}_0$ be an approximation to $A\boldsymbol{x} = \boldsymbol{b}$. Let $V \in \mathbb{C}^{n\times j}$ with $j < n$, $\operatorname{rank}(V) = j$. Define*

$$A_{\mathrm{C}} \coloneqq V^{\mathsf{H}}AV \in \mathbb{C}^{j\times j}$$

*and let further*

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + VA_{\mathrm{C}}^{-1}V^{\mathsf{H}}\boldsymbol{r}_0 \tag{2.7}$$

*with the residual $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$. Then the resulting residual $\boldsymbol{r}_1$ is orthogonal to* $\operatorname{range}(V)$*, i.e.,*

$$V^{\mathsf{H}}\boldsymbol{r}_1 = 0.$$

*In addition, if $A$ is Hermitian positive definite, then the error propagator*

$$E = I - VA_{\mathrm{C}}^{-1}V^{\mathsf{H}}A$$

*is an $A$-orthogonal projector onto the complement of* $\mathrm{range}(V)$*, i.e., the error is minimized in the $A$-norm:*

$$\|\boldsymbol{\varepsilon}_1\|_A = \|E\boldsymbol{\varepsilon}_0\|_A = \min_{\boldsymbol{y}\in\mathbb{C}^j} \|\boldsymbol{\varepsilon}_0 - V\boldsymbol{y}\|_A.$$

Since $A_{\mathrm{C}}$ is of smaller size $j < n$ than $A$, the matrix $A_{\mathrm{C}}$ is called the *coarse-grid operator* while the matrices $V$ and $V^{\mathsf{H}} = V^{\mathsf{T}}$ are called the *interpolation* and *restriction operator*, respectively, when used in a multigrid context. Note that the *coarse-grid correction* Eq. (2.7) is of the form Eq. (2.6) with $\mathfrak{I}(A)^{-1} = VA_{\mathrm{C}}^{-1}V^{\mathsf{T}}$. Note also that $\|I - VA_{\mathrm{C}}^{-1}V^{\mathsf{T}}A\|_A = 1$, i.e., the coarse-grid correction cannot increase the $A$-norm of the error.

### Coarse-grid correction in practice

Consider the error propagator $E$ of a smoother. Let us assume the eigenvalues $\lambda_i$ of $E$ (and accordingly its eigenvectors $\boldsymbol{v}_i$) are ordered such that $|\lambda_1| \geq \cdots \geq |\lambda_n|$. Let $j$ be the largest number such that $|\lambda_j| \geq \theta$ is above some threshold $\theta \approx 1$. If we choose $V$ such that $\mathrm{range}(V)$ contains all eigenvectors $\boldsymbol{v}_i$, $i = 1, \ldots, j$, with large absolute eigenvalues, then the smoother is accelerated by the coarse-grid correction Eq. (2.7). However, we assume that $A$ is large and sparse and we usually have $j = \mathcal{O}(n)$. It follows that this approach is far from practical:

- We do not know the eigenvectors $\boldsymbol{v}_i$ and calculating them would usually be prohibitively expensive.
- Even if we knew the eigenvectors, the approach would still be too expensive. Setting $V = [\boldsymbol{v}_1 \ldots \boldsymbol{v}_j]$ would generally result in a full matrix. Thus, if $j = \mathcal{O}(n)$, then the cost for storing $V$ and for matrix-vector products with $V$ would be $\mathcal{O}(n^2)$.
- Similarly, even for sparse $V$, the coarse-grid operator $A_{\mathrm{C}}$ might be dense. Its inversion would then be again too expensive.

Many algorithms to construct $V$ and $A_{\mathrm{C}}$ such that they are sparse have been published. We do not go into details and refer instead to [90, 95] for overviews.

It is worth noting, however, that the best possible convergence rate of a two-grid method is still determined by the eigenvalues of the smoother. We state this more precisely in Theorem 2.80. For this, we need the matrix $\widetilde{M} = M(M + M^{\mathsf{T}} - A)^{-1}M^{\mathsf{T}}$. It can be interpreted as the symmetrized smoother because $\widetilde{M} = \widetilde{M}^{\mathsf{T}}$ and

$$(I - M^{-\mathsf{T}}A)(I - M^{-1}A) = I - M^{-\mathsf{T}}(M + M^{\mathsf{T}} - A)M^{-1}A = I - \widetilde{M}^{-1}A.$$

Accordingly, if $M = M^{\mathsf{T}}$, then $\widetilde{M}$ corresponds to two applications of $M$. We assume that $\|I - M^{-1}A\|_A < 1$, i.e., that the smoother converges strictly monotonically in the $A$-norm.

**Theorem 2.80** ([18, Lemma 1]). *Let $V \in \mathbb{R}^{n\times j}$ be full rank. Let $\|I - M^{-1}A\|_A < 1$ and $\widetilde{M} = M(M + M^{\mathsf{T}} - A)^{-1}M^{\mathsf{T}}$. Denote by $\lambda_1 \geq \cdots \geq \lambda_n$ and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ the eigenvalues and $A$-orthogonal eigenvectors of $I - \widetilde{M}^{-1}A$. Then the minimal convergence rate of the two-grid method*

$$E(V) = (I - M^{-1}A)(I - VA_{\mathrm{C}}^{-1}V^{\mathsf{T}}A)$$

*is given by*

$$\min_V \|E(V)\|_A^2 = \lambda_{j+1}$$

*and it is obtained whenever*

$$\operatorname{range}(V) = \operatorname{span}(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_j).$$

#### Multigrid method

A multigrid method is obtained by recursing on the two-grid method: The coarse-grid correction needs to solve the linear system

$$A_{\mathrm{C}}\boldsymbol{y}_k = V^{\mathsf{T}}\boldsymbol{r}_k,$$

where $A_{\mathrm{C}}$ is sparse. This system can be solved by choosing a suitable smoother coupled with a coarse-grid correction that is defined via $A_{\mathrm{C}}$ instead of $A$. Continuing in this manner, we eventually obtain a matrix that is small enough that we can solve its linear system by a direct method. Note that this means we do not invert $A_{\mathrm{C}}$ exactly and only use an approximation $\tilde{\boldsymbol{y}}_k$ to $\boldsymbol{y}_k$. This might lead to slower convergence than anticipated from the finest two-grid method. It does not essentially alter the argument, however. For example, as long as $\tilde{\boldsymbol{y}}_k$ is close enough to $\boldsymbol{y}_k$, i.e., $\|\boldsymbol{y}_k - \tilde{\boldsymbol{y}}_k\|_A \leq \|\boldsymbol{y}_k\|_A$, the approximate coarse-grid correction still cannot increase the error $\boldsymbol{\varepsilon}_k$, see [90, Lemma A.2.2].

## 2.6. Matrix pencils

This section gives a short introduction to matrix pencils and is inspired by [CF, 87].

**Definition 2.81.** Let $T^{(0)}, T^{(1)} \in \mathbb{C}^{n\times n}$ be two matrices. Then the function $T : \mathbb{C} \to \mathbb{C}^{n\times n}$ defined by

$$T(s) = T^{(0)} - sT^{(1)}$$

is called a *(matrix) pencil*.

Notice that we can write the eigenvalue problem of a matrix $T^{(0)}$ as

$$T^{(0)}\boldsymbol{v} = \tau\boldsymbol{v} \iff (T^{(0)} - \tau I)\boldsymbol{v} = 0,$$

where $T^{(0)} - \tau I$ is a special case of a matrix pencil with $T^{(1)} = I$. Matrix pencils, therefore, give rise to *generalized eigenvalue problems*

$$(T^{(0)} - \tau T^{(1)})\boldsymbol{v} = 0 \iff T^{(0)}\boldsymbol{v} = \tau T^{(1)}\boldsymbol{v}.$$

**Definition 2.82.** A number $\tau \in \mathbb{C}$ is called an *eigenvalue* of the pencil $T(s) = T^{(0)} - sT^{(1)}$ if there is a non-zero vector $\boldsymbol{v}$ such that

$$(T^{(0)} - \tau T^{(1)})\boldsymbol{v} = 0.$$

Then $\boldsymbol{v}$ is called an *eigenvector* of $T$. The set of all eigenvalues of a pencil is called its *spectrum* $\operatorname{specp}(T)$.

**Remark 2.83.** Typically, one also says that the pencil $T$ has an eigenvalue at infinity, $\tau = \infty$, if $T^{(1)}$ is singular [87, Chapter VI]. We explicitly do not include $\infty$ in Definition 2.82.

The determinant of a pencil can be identically zero. We classify pencils accordingly:

**Definition 2.84.** Let $T(s) = T^{(0)} - sT^{(1)}$ be a matrix pencil. If $\det(T^{(0)} - sT^{(1)}) \equiv 0$ (that is $\det(T^{(0)} - sT^{(1)}) = 0$ for all values of $s$), then we call $T$ *singular*, otherwise *regular*.

We solely encounter regular pencils in this thesis. For regular pencils, the Jordan canonical form generalizes to the Weierstrass canonical form.

**Theorem 2.85** ([87, Chapter VI, Theorem 1.13], cf. [CF, Section 4.2])**.** *Let $T(s) = T^{(0)} - sT^{(1)}$ with $T^{(0)}, T^{(1)} \in \mathbb{C}^{n\times n}$ be a regular matrix pencil. Let $\tau_j$ denote the eigenvalues of $T$. Then there exist non-singular matrices $U, V \in \mathbb{C}^{n\times n}$ such that*

$$U(T^{(0)} - sT^{(1)})V = (J^{(0)} \hat{\oplus} I_{n^{(1)}}) - s(I_{n^{(0)}} \hat{\oplus} J^{(1)}) = \begin{bmatrix} J^{(0)} - sI_{n^{(0)}} & \\ & I_{n^{(1)}} - sJ^{(1)} \end{bmatrix}$$

*with the Jordan matrices*

$$J^{(0)} = \bigoplus_{j=1}^{k_0} J(\tau_j, n_j^{(0)}), \quad J^{(1)} = \bigoplus_{j=1}^{k_1} J(0, n_j^{(1)}).$$

*Here, $n^{(i)} = \sum_{j=1}^{k_i} n_j^{(i)}$ for $i = 0, 1$ and $n^{(0)} + n^{(1)} = n$. $J(\cdot,\cdot)$ denotes a Jordan block (see Definition 2.50). This decomposition is known as the* Weierstrass canonical form*.*

We know that a Hermitian matrix can be unitarily diagonalized. For pencils, a similar result holds:

**Theorem 2.86** ([87, Chapter VI, Theorem 1.15])**.** *Let $T^{(0)} \in \mathbb{C}^{n\times n}$ be Hermitian and $T^{(1)} \in \mathbb{C}^{n\times n}$ be Hermitian positive definite. Then the eigenvalues $\tau_i$ of the pencil $T^{(0)} - sT^{(1)}$ are real and its eigenvectors $\boldsymbol{v}_i$ can be chosen to be $T^{(1)}$-orthogonal. That is, there exists a non-singular matrix $Q = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n]$ and a real diagonal matrix $D = \operatorname{diag}(\tau_i)$ satisfying*

$$T^{(0)}Q = T^{(1)}QD$$

*and*

$$Q^{\mathsf{H}}T^{(1)}Q = I = QQ^{\mathsf{H}}T^{(1)}.$$

# 3. CF-matrices

In this chapter, we develop a new way of expressing the action of rational matrix functions $r(A)\boldsymbol{b}$ based on continued fractions and then examine if this new representation offers any computational advantage. Of course, we compare it to what we discussed in Section 2.5, i.e., to the linear shifted systems $(A - \tau_j I)^{-1}\boldsymbol{b}$ (generated by the partial fraction expansion) to which we apply a Krylov subspace method or a multigrid method.

For the idea presented here, we were inspired by work [17, 75] in the context of "Quantum Chromodynamics" (QCD), i.e., theoretical physics, where the sign function of a matrix is approximated using continued fractions.

We have already published a large part of the results in this chapter in [CF]. However, many parts are new or have been completely rewritten. We summarize the major changes: We slightly changed the representation by starting from a continued fraction for $r(s)$ instead of one for $r(s)^{-1}$, see Remark 3.12 for a comparison. This simplifies the resulting matrices and our proofs. We included a new subsection about Krylov subspace methods (Section 3.2.3). The proofs in Section 3.2.4 were simplified (Theorem 3.21) or modified to yield a stronger result (Theorem 3.24). The numerical experiments in Section 3.3 were adapted to fit the rest of the chapter. In particular, we present new experiments regarding multigrid methods in Section 3.3.3.

## 3.1. Introduction

The main concept discussed in this section is the *CF-matrix*, which enables us to represent the action of a rational matrix function by a single matrix inverse.

### 3.1.1. Basic properties

We again consider continued fractions as rational functions as in Section 2.2. Let $g_m$ be the finite continued fraction

$$g_m(s) = b_0 + \mathop{\mathrm{K}}_{i=1}^{m} \left( \frac{c_i(s)}{b_i(s)} \right).$$

Let $c_i$ and $b_i$ be polynomials of degree at most $\ell$ so we can write

$$b_i(s) = \sum_{j=0}^{\ell} b_i^{(j)} s^j, \quad c_i(s) = \sum_{j=0}^{\ell} c_i^{(j)} s^j.$$

On the other hand, let us define the tridiagonal matrix

$$T_m(s) = \begin{bmatrix} \beta_1(s) & \gamma_2(s) & & \\ \alpha_2(s) & \beta_2(s) & \ddots & \\ & \ddots & \ddots & \gamma_m(s) \\ & & \alpha_m(s) & \beta_m(s) \end{bmatrix} \in \mathbb{C}^{m\times m},$$

with polynomials $\alpha_i$, $\beta_i$, $\gamma_i$. We know from Theorem 2.42 that we can express a (constant) finite continued fraction via the inverse of a tridiagonal matrix. The same holds for continued fractions that contain polynomials. In particular,

$$g_m(s) = b_0(s) + c_1(s)\boldsymbol{e}_1^{\mathsf{T}} T_m(s)^{-1}\boldsymbol{e}_1,$$

if $g_m(s)$ is defined and

$$\begin{aligned} \beta_i(s) &= b_i(s), \quad i = 1,\dots,m, \\ -\alpha_i(s)\gamma_i(s) &= c_i(s), \quad i = 2,\dots,m. \end{aligned} \tag{3.1}$$

Now we want to evaluate the continued fraction at the matrix $A$. If we interpret $g_m(s) = r(s) = \frac{p(s)}{q(s)}$ as a rational function, we can write $r(A) = g_m(A)$. The definition of continued fractions extends easily to matrices:

$$\begin{aligned} g_m(A) &= b_0(A) + c_1(A)\Big(b_1(A) + c_2(A)\big(b_2(A) + \dots\big)^{-1}\Big)^{-1} \\ &= p(A)q(A)^{-1} = r(A). \end{aligned} \tag{3.2}$$

How does this translate to the matrix $T_m(s)$? Its elements are polynomials in $s$, so $T_m(A)$ should be a matrix with polynomials in $A$ as its elements.

**Definition 3.1.** Let $A \in \mathbb{C}^{n\times n}$ be a matrix and

$$g_m(s) = \operatorname*{K}_{i=1}^{m} \left(\frac{c_i(s)}{b_i(s)}\right)$$

be a finite continued fraction that is defined on the spectrum of $A$. Let $\alpha_i(s)$, $\beta_i(s)$, $\gamma_i(s)$ be polynomials that fulfill Eq. (3.1). Then the *CF-matrix* of $g_m(s)$ and $A$ is defined as the block tridiagonal matrix

$$T_m(A) = \begin{bmatrix} \beta_1(A) & \gamma_2(A) & & \\ \alpha_2(A) & \beta_2(A) & \ddots & \\ & \ddots & \ddots & \gamma_m(A) \\ & & \alpha_m(A) & \beta_m(A) \end{bmatrix} \in \mathbb{C}^{nm\times nm}.$$

Having defined $T_m(A)$, we now show how to obtain $g_m(A)$ from its inverse $T_m(A)^{-1}$. For this, we use the block UDL decomposition. Compared to the related LDU decomposition, the UDL decomposition is somewhat uncommon but it is by no means a new concept, see, e.g., [35].

**Lemma 3.2** (cf. [CF, Lemma 3.6])**.** *Let $T_m(A)$ be a CF-matrix. Let all matrices $\Sigma_i$ defined by*

$$\Sigma_m = \beta_m(A), \quad \Sigma_i = \beta_i(A) - \gamma_{i+1}(A)\Sigma_{i+1}^{-1}\alpha_i(A), \quad i = m-1, \ldots, 1,$$

*be non-singular. Then $T_m(A)$ has the block UDL decomposition*

$$T_m(A) = UDL,$$

*where $D = \operatorname{diag}(\Sigma_1, \ldots, \Sigma_m)$ and*

$$U = \begin{bmatrix} I_n & \gamma_2(A)\Sigma_2^{-1} & & \\ & \ddots & \ddots & \\ & & I_n & \gamma_m(A)\Sigma_m^{-1} \\ & & & I_n \end{bmatrix}, \quad L = \begin{bmatrix} I_n & & & \\ \Sigma_2^{-1}\alpha_2(A) & I_n & & \\ & \ddots & \ddots & \\ & & \Sigma_m^{-1}\alpha_m(A) & I_n \end{bmatrix}.$$

*Proof.* This is easily verified by direct calculation. □

**Theorem 3.3** (cf. [CF, Theorem 3.7])**.** *Let $T_m(A)$ be the non-singular CF-matrix of $g_m(s)$ and $A \in \mathbb{C}^{n\times n}$. Let $r(s) = g_m(s)$ be defined on the spectrum of $A$. Then*

$$g_m(A) = b_0(A) + c_1(A)(\boldsymbol{e}_1^{\mathsf{T}} \otimes I_n)T_m(A)^{-1}(\boldsymbol{e}_1 \otimes I_n),$$

*where $\otimes$ denotes the Kronecker product and $\boldsymbol{e}_1$ is the first unit vector in $\mathbb{C}^m$.*

*Proof.* We first assume that all $\Sigma_i$ of the UDL decomposition of $T_m(A)$ (Lemma 3.2) are non-singular. Then we have

$$T_m(A)^{-1} = (UDL)^{-1} = L^{-1}D^{-1}U^{-1}.$$

Because of the block structure of $L$ and $U$, one quickly verifies that

$$\begin{aligned} U(\boldsymbol{e}_1 \otimes I_n) = \boldsymbol{e}_1 \otimes I_n \implies U^{-1}(\boldsymbol{e}_1 \otimes I_n) = \boldsymbol{e}_1 \otimes I_n, \\ (\boldsymbol{e}_1^{\mathsf{T}} \otimes I_n)L = \boldsymbol{e}_1^{\mathsf{T}} \otimes I_n \implies (\boldsymbol{e}_1^{\mathsf{T}} \otimes I_n)L^{-1} = \boldsymbol{e}_1^{\mathsf{T}} \otimes I_n. \end{aligned}$$

From this, we see that

$$(\boldsymbol{e}_1^{\mathsf{T}} \otimes I_n)T_m(A)^{-1}(\boldsymbol{e}_1 \otimes I_n) = (\boldsymbol{e}_1^{\mathsf{T}} \otimes I_n)D^{-1}(\boldsymbol{e}_1 \otimes I_n) = \Sigma_1^{-1}.$$

The matrices $\Sigma_i$ can be interpreted as rational functions in $A$. Matrix functions in the same matrix commute with each other (see Section 2.3), so we have by construction

$$\Sigma_1 = \beta_1(A) - \gamma_2(A)\Sigma_2^{-1}\alpha_2(A) = b_1(A) - c_2(A)\Sigma_2^{-1}.$$

Continuing similarly for the other $\Sigma_i$, $i = 2, \ldots, m$, we see that

$$g_m(A) = b_0(A) + c_1(A)\Sigma_1^{-1} = b_0(A) + c_1(A)(\boldsymbol{e}_1^\mathsf{T} \otimes I_n)T_m(A)^{-1}(\boldsymbol{e}_1 \otimes I_n),$$

cf. Eq. (3.2).

If any $\Sigma_i$ is singular, we use a continuity argument: Let $A_\epsilon = A + \epsilon I$ with $\epsilon > 0$. It is clear that $\operatorname{spec}(A_\epsilon) = \operatorname{spec}(A) + \epsilon$. A rational function is holomorphic everywhere except for a finite set of points (its poles). As $g_m(s)$ is a rational function defined on the spectrum of $A$, $g_m(A_\epsilon)$ is also defined for small enough $\epsilon$ and

$$g_m(A) = \lim_{\epsilon \to 0} g_m(A_\epsilon).$$

Similarly, even if $\Sigma_i = r_i(A)$ is singular for some $i$, there are small enough $\epsilon$ such that all $r_i(A_\epsilon)$ are non-singular because $\operatorname{spec}(A_\epsilon)$ and the set of poles of all $r_i(s)$ are both finite sets. For these $\epsilon$, we know from the above that

$$g_m(A_\epsilon) = b_0(A_\epsilon) + c_1(A_\epsilon)(\boldsymbol{e}_1^\mathsf{T} \otimes I_n)T_m(A_\epsilon)^{-1}(\boldsymbol{e}_1 \otimes I_n)$$

and it follows

$$g_m(A) = \lim_{\epsilon \to 0} g_m(A_\epsilon) = b_0(A) + c_1(A)(\boldsymbol{e}_1^\mathsf{T} \otimes I_n)T_m(A)^{-1}(\boldsymbol{e}_1 \otimes I_n). \qquad \square$$

**Corollary 3.4.** *Let $g_m(s) = r(s)$ be a continued fraction and let $T_m(A)$ be the non-singular CF-matrix of $g_m(s)$ and $A$. Then*

$$r(A)\boldsymbol{b} = b_0(A)\boldsymbol{b} + c_1(A)(\boldsymbol{e}_1^\mathsf{T} \otimes I_n)T_m(A)^{-1}(\boldsymbol{e}_1 \otimes \boldsymbol{b}).$$

We saw in Corollary 2.45 that the representation of $g_m$ by the tridiagonal matrix $T_m$ remains valid after multiplying the latter by matrices that have $\boldsymbol{e}_1$ as eigenvector. This easily extends to CF-matrices:

**Corollary 3.5.** *Let $\lambda \neq 0$ and $H_\ell, H_r \in \mathbb{C}^{m\times m}$ be matrices such that*

$$H_\ell^{-1}\boldsymbol{e}_1 = \lambda \boldsymbol{e}_1,$$
$$\boldsymbol{e}_1^\mathsf{T} H_r^{-1} = \lambda^{-1}\boldsymbol{e}_1^\mathsf{T}.$$

*Then Theorem 3.3 still holds if $T_m(A)$ is multiplied by $H_\ell \otimes I$ and $H_r \otimes I$ from the left and the right, respectively. That is*

$$g_m(A) = b_0(A) + c_1(A)(\boldsymbol{e}_1^\mathsf{T} \otimes I_n)\widetilde{T}_m(A)^{-1}(\boldsymbol{e}_1 \otimes I_n)$$

*with*

$$\widetilde{T}_m(A) = (H_\ell \otimes I)T_m(A)(H_r \otimes I).$$

We now express $T_m(A)$ as a sum of Kronecker products, which will help further investigations. Let us shortly return to the scalar case $T_m(s)$. We know that $\beta_i^{(j)} = b_i^{(j)}$ and that the degrees of all $\alpha_i$ and $\gamma_i$ are bounded by $\ell$. We can thus write $T_m(s)$ as a polynomial with matrix coefficients. To this end, we define the tridiagonal matrices

$$T_m^{(j)} = \begin{bmatrix} \beta_1^{(j)} & \gamma_2^{(j)} & & \\ \alpha_2^{(j)} & \beta_2^{(j)} & \ddots & \\ & \ddots & \ddots & \gamma_m^{(j)} \\ & & \alpha_m^{(j)} & \beta_m^{(j)} \end{bmatrix} \in \mathbb{C}^{m\times m}, \quad j = 0, \dots, \ell, \tag{3.3}$$

where the elements are the coefficients of the polynomials

$$\alpha_i(s) = \sum_{j=0}^{\ell} \alpha_i^{(j)} s^j, \quad \beta_i(s) = \sum_{j=0}^{\ell} \beta_i^{(j)} s^j, \quad \gamma_i(s) = \sum_{j=0}^{\ell} \gamma_i^{(j)} s^j. \tag{3.4}$$

Consequently, we can write

$$T_m(s) = \sum_{j=0}^{\ell} T_m^{(j)} s^j.$$

We extend this idea to the matrix case:

**Lemma 3.6.** *Let $T_m^{(j)}$ be defined as in Eq.* (3.3) *with Eq.* (3.4)*. The CF-matrix $T_m(A)$ can be written as the sum of Kronecker products, i.e.,*

$$T_m(A) = \sum_{j=0}^{\ell} T_m^{(j)} \otimes A^j.$$

*Proof.* This follows immediately from

$$\begin{aligned} T_m(A) &= \begin{bmatrix} \sum_{j=0}^{\ell} \beta_1^{(j)} A^j & \sum_{j=0}^{\ell} \gamma_2^{(j)} A^j & & \\ \sum_{j=0}^{\ell} \alpha_2^{(j)} A^j & \sum_{j=0}^{\ell} \beta_2^{(j)} A^j & \ddots & \\ & \ddots & \ddots & \sum_{j=0}^{\ell} \gamma_m^{(j)} A^j \\ & & \sum_{j=0}^{\ell} \alpha_m^{(j)} A^j & \sum_{j=0}^{\ell} \beta_m^{(j)} A^j \end{bmatrix} \\ &= \sum_{j=0}^{\ell} \begin{bmatrix} \beta_1^{(j)} A^j & \gamma_2^{(j)} A^j & & \\ \alpha_2^{(j)} A^j & \beta_2^{(j)} A^j & \ddots & \\ & \ddots & \ddots & \gamma_m^{(j)} A^j \\ & & \alpha_m^{(j)} A^j & \beta_m^{(j)} A^j \end{bmatrix} = \sum_{j=0}^{\ell} T_m^{(j)} \otimes A^j. \end{aligned}$$

□

Note that $-\alpha_i(s)\gamma_i(s) \equiv c_i(s)$ does not imply that $-\alpha_i^{(j)}\gamma_i^{(j)} = c_i^{(j)}$, thus the matrices $T_m(s)$ and $T_m^{(j)}$ are not unique for a given continued fraction $g_m(s)$, cf. Corollary 2.45. Moreover, it turns out that most of the following holds for any matrix $T(A)$ of the form $\sum_{j=0}^{\ell} T^{(j)} \otimes A^j$, i.e., for arbitrary $T^{(j)}$ that are not necessarily connected to a known continued fraction. Because of this and for the sake of notational simplicity, we use an index $m$ in the following only if we want to emphasize that the matrix is constructed from a continued fraction $g_m$.

The representation with Kronecker products allows us to connect the spectrum of $T(A)$ and the spectrum of $T(s)$ for particular $s$.

**Lemma 3.7.** *Let $T(A) = \sum_{j=0}^{\ell} T^{(j)} \otimes A^j$ with $T^{(j)} \in \mathbb{C}^{m\times m}$ and $A \in \mathbb{C}^{n\times n}$. Let $\lambda_i$ for $i = 1, \ldots, n$ be the eigenvalues of $A$. Then*

$$\operatorname{spec}(T(A)) = \bigcup_{i=1}^{n} \operatorname{spec}(T(\lambda_i)).$$

*Proof.* Let $A = ZJZ^{-1}$ be the Jordan canonical form of $A$. As similarity transformations do not change the eigenvalues, we have

$$\operatorname{spec}(T(A)) = \operatorname{spec}\big((I \otimes Z)^{-1} T(A)(I \otimes Z)\big) = \operatorname{spec}\left(\sum_{j=0}^{\ell} T^{(j)} \otimes J^j\right).$$

It is known that there exists a (perfect shuffle) permutation matrix $P$ such that

$$A \otimes B = P(B \otimes A)P^{-1}$$

for any square matrices $A$ and $B$, see [44, Eq. (1.3.5)] or [57, Eq. (4.3.12)]. Applying this similarity transformation, we have

$$\operatorname{spec}\left(\sum_{j=0}^{\ell} T^{(j)} \otimes J^j\right) = \operatorname{spec}\left(\sum_{j=0}^{\ell} J^j \otimes T^{(j)}\right).$$

Note that $J^j \otimes T^{(j)}$ is block upper triangular. The eigenvalues of such matrices are those of their diagonal blocks:

$$\operatorname{spec}\left(\sum_{j=0}^{\ell} J^j \otimes T^{(j)}\right) = \bigcup_{i=1}^{n} \operatorname{spec}\left(\sum_{j=0}^{\ell} (J^j)_{i,i} T^{(j)}\right).$$

The diagonal entries of $J^j$ are just $\lambda_i^j$. This means $\sum_{j=0}^{\ell} (J^j)_{i,i} T^{(j)} = T(\lambda_i)$. □

In Theorem 3.3, we had to assume that $T_m(A)$ is non-singular. We use Lemma 3.7 to show that it is sufficient to have the rational function be defined on the spectrum of $A$.

**Corollary 3.8.** *The CF-matrix $T_m(A)$ of $g_m(s)$ and $A$ is non-singular if $r(s) = g_m(s)$ is defined on the spectrum of $A$.*

*Proof.* As $r(\lambda)$ is defined for all eigenvalues $\lambda$ by hypothesis, we know $\det(T_m(\lambda)) \neq 0$ by Theorem 2.42 and thus $\det(T_m(A)) \neq 0$ by Lemma 3.7. □

### 3.1.2. Construction

We saw that we can write any matrix continued fraction as

$$g_m(A) = b_0(A) + c_1(A)(\boldsymbol{e}_1^\mathsf{T} \otimes I_n)T_m(A)^{-1}(\boldsymbol{e}_1 \otimes I_n)$$

with $T_m(A) = \sum_{j=0}^{\ell} T_m^{(j)} \otimes A^j$ for some $\ell$. We now give some examples (similarly to [CF, Section 4.1]) for the structure of $T_m(A)$ for some kinds of continued fractions. These examples illustrate that often we only need the first two terms $T_m^{(0)}$ and $T_m^{(1)}$, which is advantageous: While we might not need to store $T_m(A)$ explicitly, we generally want to avoid higher powers of $A$ even implicitly.[1]

**Example 3.9** (Regular C-fractions)**.** In the approximant of a regular C-fraction (Definition 2.47)

$$g_m(s) = b_0 + \mathop{\mathrm{K}}_{i=1}^{m} \left(\frac{c_i s}{1}\right),$$

the partial denominators are all 1. Because of this, we have $\beta_i(s) = 1$. For the partial numerators, we can choose $\alpha_i(s) = -c_i s$ and $\gamma_i(s) = 1$ so that $-\alpha_i(s)\gamma_i(s) = c_i s$. This results in

$$T_m(s) = T_m^{(0)} - T_m^{(1)} s \implies T_m(A) = T_m^{(0)} \otimes I - T_m^{(1)} \otimes A$$

with

$$T_m^{(0)} = \begin{bmatrix} 1 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & 1 \end{bmatrix}, \quad T_m^{(1)} = \begin{bmatrix} 0 & & & \\ c_2 & \ddots & & \\ & \ddots & \ddots & \\ & & c_m & 0 \end{bmatrix}.$$

Here, we chose the subdiagonal to contain the coefficients $c_i$, but we could as well choose the superdiagonal, i.e., we could take the pair $(T_m^{(0)})^\mathsf{T}$, $(T_m^{(1)})^\mathsf{T}$ instead of $T_m^{(0)}$, $T_m^{(1)}$.

Since Padé approximations $r_{k,k'}(s)$ can often be represented by C-fractions (Theorem 2.48), we can represent $r_{k,k'}(A)$ by a CF-matrix $T_m(A)$ in this way. However, increasing $m$ increases only either the degree of the numerator $k$ or the degree of the denominator $k'$. Using a contraction, we can increase both at the same time but still only need the first two terms $T_m^{(0)}$, $T_m^{(1)}$ for $T_m(A)$.

**Example 3.10** (Contracted regular C-fractions)**.** We use the contraction Lemma 2.46 of the regular C-fraction before constructing the CF-matrix (cf. Corollary 2.49). The approximants are

$$\tilde{g}_m(s) = b_0 + \frac{c_1 s}{1 + c_2 s + \mathrm{K}_{i=2}^{m} \left(\frac{-c_{2i-2}c_{2i-1}s^2}{1+(c_{2i}+c_{2i-1})s}\right)}.$$

[1] This follows a similar argument as the comparison between expressing a rational function as the quotient of two polynomials or via the partial fraction expansion in Section 2.5.

We choose $\alpha_i(s) = \sqrt{c_{2i-2}c_{2i-1}}s = \gamma_i(s)$. Then we have

$$T_m(s) = T_m^{(0)} - T_m^{(1)}s \implies T_m(A) = T_m^{(0)} \otimes I - T_m^{(1)} \otimes A$$

with

$$T_m^{(0)} = I_m,$$

$$T_m^{(1)} = (-1) \cdot \begin{bmatrix} c_2 & \sqrt{c_2c_3} & & & \\ \sqrt{c_2c_3} & c_3+c_4 & \sqrt{c_4c_5} & & \\ & \sqrt{c_4c_5} & c_5+c_6 & \cdots & \\ & & \ddots & \ddots & \sqrt{c_{2m-2}c_{2m-1}} \\ & & & \sqrt{c_{2m-2}c_{2m-1}} & c_{2m-1}+c_{2m} \end{bmatrix}.$$

Thus, if $c_{2i-2}c_{2i-1} > 0$ for $i \geq 2$, both $T_m^{(0)}$ and $T_m^{(1)}$ are symmetric, so $T_m(A)$ is symmetric if $A$ is symmetric. Another possible choice is $\alpha_i(s) = c_{2i-1}s$ and $\gamma_i(s) = c_{2i-2}s$ that does not yield symmetric matrices but avoids the introduction of imaginary numbers if $c_{2i-2}c_{2i-1} < 0$ but $c_i \in \mathbb{R}$:

$$\widetilde{T}_m^{(0)} = I_m,$$

$$\widetilde{T}_m^{(1)} = (-1) \cdot \begin{bmatrix} c_2 & c_2 & & & \\ c_3 & c_3+c_4 & c_4 & & \\ & c_5 & c_5+c_6 & \cdots & \\ & & \ddots & \ddots & c_{2m-2} \\ & & & c_{2m-1} & c_{2m-1}+c_{2m} \end{bmatrix}.$$

Note that the $T_m^{(1)}$ and $\widetilde{T}_m^{(1)}$ are similar to each other: We define $D = \operatorname{diag}(d_1, \ldots, d_m)$ with

$$d_1 = 1, \quad d_{i+1} = \sqrt{\frac{c_{2i}}{c_{2i+1}}}d_i, \quad i = 1, \ldots, m-1.$$

Then we verify that

$$D\widetilde{T}_m^{(1)}D^{-1} = T_m^{(1)}, \quad (D \otimes I)\widetilde{T}_m(A)(D^{-1} \otimes I) = T_m(A).$$

**Example 3.11** (Continued fractions by the Euclidean algorithm)**.** In Eq. (2.2), we saw that we can write every rational function as a continued fraction by using the Euclidean algorithm:

$$r(s) = \frac{p(s)}{q(s)} = b_0(s) + \operatorname*{K}_{i=1}^{m} \left( \frac{1}{b_i(s)} \right).$$

Assume that the degree of $b_i(s)$ is at most 1 for every $i = 1, \ldots, m$, i.e., $b_i(s) = b_i^{(0)} - b_i^{(1)}s$. Then

$$T_m(s) = T_m^{(0)} - T_m^{(1)}s \implies T_m(A) = T_m^{(0)} \otimes I - T_m^{(1)} \otimes A$$

with

$$T_m^{(0)} = \begin{bmatrix} b_1^{(0)} & -1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & 1 & b_n^{(0)} \end{bmatrix}, \quad T_m^{(1)} = \begin{bmatrix} b_1^{(1)} & & \\ & \ddots & \\ & & b_n^{(1)} \end{bmatrix}.$$

By Corollary 3.5, we can modify this CF-matrix to be symmetric if $A$ is symmetric. For example, we define

$$\begin{aligned} D_{\mathrm{L}} &= \operatorname{diag}((-1)^{\lfloor 0/2 \rfloor}, \dots, (-1)^{\lfloor n/2 \rfloor}), \\ D_{\mathrm{R}} &= \operatorname{diag}((-1)^{\lfloor (0+1)/2 \rfloor}, \dots, (-1)^{\lfloor (n+1)/2 \rfloor}). \end{aligned}$$

We have $D_{\mathrm{L}}\boldsymbol{e}_1 = \boldsymbol{e}_1$ and $\boldsymbol{e}_1^{\mathsf{T}} D_{\mathrm{R}} = \boldsymbol{e}_1^{\mathsf{T}}$, so Corollary 3.5 is applicable. Thus, we can use

$$D_{\mathrm{L}} T_m^{(0)} D_{\mathrm{R}} = \begin{bmatrix} (-1)^0 b_1^{(0)} & 1 & & \\ 1 & (-1)^1 b_2^{(0)} & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & (-1)^{n-1} b_n^{(0)} \end{bmatrix},$$

$$D_{\mathrm{L}} T_m^{(1)} D_{\mathrm{R}} = \operatorname{diag}((-1)^0 b_1^{(1)}, \dots, (-1)^{n-1} b_n^{(1)})$$

or, alternatively,

$$D_{\mathrm{R}} T_m^{(0)} D_{\mathrm{L}} = \begin{bmatrix} (-1)^0 b_1^{(0)} & -1 & & \\ -1 & (-1)^1 b_2^{(0)} & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & (-1)^{n-1} b_n^{(0)} \end{bmatrix},$$

$$D_{\mathrm{R}} T_m^{(1)} D_{\mathrm{L}} = D_{\mathrm{L}} T_m^{(1)} D_{\mathrm{R}}.$$

Strictly speaking, we cannot call $(D_{\mathrm{L}} \otimes I) T_m(A) (D_{\mathrm{R}} \otimes I)$ a CF-matrix of $r(s)$ because every second diagonal entry of $D_{\mathrm{L}} T_m(s) D_{\mathrm{R}}$ has the wrong sign. However, $(D_{\mathrm{L}} \otimes I) T_m(A) (D_{\mathrm{R}} \otimes I)$ is still a CF-matrix of an expanded continued fraction that is equivalent to $r(s)$, see Lemma 2.44 and Corollary 3.5.

**Remark 3.12.** We have been constructing CF-matrices such that the terms $b_0(A)$ and $c_1(A)$ need to be evaluated separately. They can be included by noting that

$$g_m(s) = b_0(s) + \operatorname*{K}_{i=1}^{m} \left( \frac{c_i(s)}{b_i(s)} \right) = \cfrac{1}{0 + \cfrac{1}{b_0(s) + \mathrm{K}_{i=1}^{m} \left( \frac{c_i(s)}{b_i(s)} \right)}}.$$

Accordingly, we can write

$$g_m(A) = (\boldsymbol{e}_1^\mathsf{T} \otimes I)\widetilde{T}_{m+2}(A)^{-1}(\boldsymbol{e}_1 \otimes I)$$

with, e.g.,

$$\widetilde{T}_{m+2}(A) = \begin{bmatrix} 0 & I & & & & & \\ -I & b_0(A) & c_1(A) & & & & \\ & -I & b_1(A) & c_2(A) & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -I & b_{m-1}(A) & c_{m-1}(A) \\ & & & & -I & b_m(A) \end{bmatrix} \in \mathbb{C}^{(m+2)n\times(m+2)n}.$$

Similarly, assume we have the inverse of a rational function given as a continued fraction,

$$r(s) = \frac{1}{g_m(s)}.$$

Then

$$r(A) = g_m(A)^{-1} = (\boldsymbol{e}_1^\mathsf{T} \otimes I)T_{m+1}(A)^{-1}(\boldsymbol{e}_1 \otimes I)$$

with

$$T_{m+1}(A) = \begin{bmatrix} b_0(A) & c_1(A) & & & \\ -I & b_1(A) & c_2(A) & & \\ & \ddots & \ddots & \ddots & \\ & & -I & b_{m-1}(A) & c_{m-1}(A) \\ & & & -I & b_m(A) \end{bmatrix} \in \mathbb{C}^{(m+1)n\times(m+1)n}.$$

We took this approach in [CF]. Note that it increases the size of the CF-matrices by one block. Moreover, for contracted regular C-fractions, $T^{(1)}_{m+1}$ has a zero on its subdiagonal, i.e., we cannot symmetrize the matrix as we did before.

## 3.2. Search for numerical methods

We established that we can evaluate the action $r(A)\boldsymbol{b}$ of a rational matrix function $r(A) = g_m(A)$ by

$$r(A)\boldsymbol{b} = b_0(A)\boldsymbol{b} + c_1(A)(\boldsymbol{e}_1^\mathsf{T} \otimes I)T_m(A)^{-1}(\boldsymbol{e}_1 \otimes \boldsymbol{b})$$

and what the matrix $T_m(A)$ can look like. The evaluation of $p(A)\boldsymbol{b}$, where $p$ is a polynomial, is straightforward. This is why we need to investigate only the step $(\boldsymbol{e}_1^\mathsf{T} \otimes I)T_m(A)^{-1}(\boldsymbol{e}_1 \otimes \boldsymbol{b})$. In this section, we investigate numerical methods for solving this linear system, i.e., for

$$T_m(A)\boldsymbol{x} = \boldsymbol{e}_1 \otimes \boldsymbol{b}.$$

Further, we assume that

$$T_m(A) = T_m^{(0)} \otimes I - T_m^{(1)} \otimes A.$$

As was shown in Section 3.1.2, this restriction covers, e.g., regular C-fractions and thus Padé approximations (recall Theorem 2.48). Consequently, we retain many interesting rational functions. In addition, higher powers of $A$ would increase the number of non-zero elements and the cost of matrix-vector products, both of which one wants to avoid for iterative methods. Furthermore, $T_m(s) = T_m^{(0)} - sT_m^{(1)}$ is now a pencil (see Section 2.6). In our analysis, we use its Weierstrass canonical form (Theorem 2.85), which requires the pencil to be regular. This is, however, not a restriction:

**Lemma 3.13.** *Let $T_m(s) = T_m^{(0)} - sT_m^{(1)}$ be a pencil and the rational function $r(s) = \boldsymbol{e}_1^\mathsf{T} T_m(s)^{-1} \boldsymbol{e}_1$ be defined on the spectrum of $A$. Then $T_m(s)$ is regular.*

*Proof.* As $r(s)$ is defined on the spectrum of $A$, we have $\det(T_m(\lambda)) \neq 0$ for any eigenvalue $\lambda$ of $A$. This means $\det(T_m(s)) \not\equiv 0$. □

When constructing numerical methods involving a matrix, one should try to exploit as much structural information about it as possible. This is especially true in our case because $T_m(A)$ is larger than $A$ by a factor of $m$. This is why our analysis focuses on the assumption that we want to exploit the Kronecker product structure of $T_m(A)$. Of course, we could instead just apply a black-box method to $T_m(A)\boldsymbol{x} = \boldsymbol{c}$ (where $\boldsymbol{c} = \boldsymbol{e}_1 \otimes \boldsymbol{b}$), e.g., a generic algebraic multigrid method. We cannot rule out that in some cases this would yield some advantage compared to more refined approaches. On the other hand, we also have no reason to believe this. This is all the more true if the partial fraction expansion of $g_m(s)$ is available as it is usually easier to work with several small matrices (like $A - \tau_j I$) instead of a single large one (like $T_m(A)$).

As before, we write the index $m$ only if we want to emphasize that the matrix is constructed from a continued fraction. For a generic pencil $T(s) = T^{(0)} - sT^{(1)}$, the matrix $T(A)$ then denotes evidently $T(A) = T^{(0)} \otimes I - T^{(1)} \otimes A$.

### 3.2.1. Partial fraction expansion

As we have discussed in Section 2.5, we can efficiently evaluate $r(A)\boldsymbol{b}$ using the partial fraction expansion:

$$r(A)\boldsymbol{b} = p_0(A)\boldsymbol{b} + \sum_{j=1}^{k} \sum_{i=1}^{m_j} w_{j,i}(A - \tau_j I)^{-i}\boldsymbol{b}. \tag{3.5}$$

We also know that $r(A)\boldsymbol{b} = (\boldsymbol{e}_1^\mathsf{T} \otimes I) T_m(A)^{-1}(\boldsymbol{e}_1 \otimes \boldsymbol{b})$, which implies that

$$(\boldsymbol{e}_1^\mathsf{T} \otimes I) T_m(A)^{-1}(\boldsymbol{e}_1 \otimes \boldsymbol{b}) = p_0(A)\boldsymbol{b} + \sum_{j=1}^{k} \sum_{i=1}^{m_j} w_{j,i}(A - \tau_j I)^{-i}\boldsymbol{b}.$$

In cases where both the partial fraction expansion and a continued fraction expansion of a rational function are known, the two approaches thus are in direct competition. We now show how the partial fraction expansion can be retrieved from any regular pencil.

**Theorem 3.14** (cf. [CF, Theorem 4.1] and [14, Theorem 3.1])**.** *Let $T(s) = T^{(0)} - sT^{(1)}$ be a regular pencil. Let $U$, $V$ and $J(\tau_j, m_j^{(0)})$, $J(0, m_j^{(1)})$ be the matrices and parameters of its Weierstrass canonical form, i.e.,*

$$UT^{(0)}V = J^{(0)} \mathbin{\hat{\oplus}} I_{m^{(1)}}, \quad UT^{(1)}V = I_{m^{(0)}} \mathbin{\hat{\oplus}} J^{(1)}$$

*with the Jordan matrices*

$$J^{(0)} = \bigoplus_{j=1}^{k_0} J(\tau_j, m_j^{(0)}), \quad J^{(1)} = \bigoplus_{j=1}^{k_1} J(0, m_j^{(1)}).$$

*Let $\boldsymbol{u} = U\boldsymbol{e}_1$ and $\boldsymbol{v}^\mathsf{T} = \boldsymbol{e}_1^\mathsf{T} V$. Denote by $\boldsymbol{u}^{(j)}$ and $\boldsymbol{v}^{(j)}$, $j = 1, \ldots, k_0 + k_1$, their blocks corresponding to block $j$ of the Weierstrass canonical form. Define*

$$w_{j,i} = -(\boldsymbol{v}^{(j)})^\mathsf{T} \big(S_{m_j^{(0)}}\big)^{i-1} \boldsymbol{u}^{(j)},$$
$$\sigma_{j,i} = (\boldsymbol{v}^{(k_0+j)})^\mathsf{T} \big(S_{m_j^{(1)}}\big)^{i} \boldsymbol{u}^{(k_0+j)}$$

*with the matrices*

$$S_m = \begin{bmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & 0 \end{bmatrix} \in \mathbb{C}^{m \times m}$$

*from Definition 2.50. Then*

$$\boldsymbol{e}_1^\mathsf{T} T(s)^{-1} \boldsymbol{e}_1 = \sum_{j=1}^{k_0} \sum_{i=1}^{m_j^{(0)}} \frac{w_{j,i}}{(s-\tau_j)^i} + \sum_{j=1}^{k_1} \sum_{i=0}^{m_j^{(1)}-1} s^i \sigma_{j,i}.$$

*Proof.* The Weierstrass canonical form implies

$$T(s)^{-1} = V\big((J^{(0)} - sI) \mathbin{\hat{\oplus}} (-sJ^{(1)} + I)\big)^{-1} U,$$

so we have

$$\boldsymbol{e}_1^\mathsf{T} T(s)^{-1} \boldsymbol{e}_1 = \boldsymbol{v}^\mathsf{T} \big((J^{(0)} - sI)^{-1} \mathbin{\hat{\oplus}} (-sJ^{(1)} + I)^{-1}\big) \boldsymbol{u}. \tag{3.6}$$

For any Jordan block $J(\mu, m) = \mu I + S_m$, we know that

$$J(\mu, m)^{-1} = \sum_{i=0}^{m-1} \frac{(-1)^i}{\mu^{i+1}} \big(S_m\big)^i = \sum_{i=0}^{m-1} \frac{-1}{(-\mu)^{i+1}} \big(S_m\big)^i,$$

cf. Definition 2.52. As $J(\mu, m) - sI = J(\mu - s, m)$, this gives

$$(J^{(0)} - sI)^{-1} = \bigoplus_{j=1}^{\widehat{k_0}} J(\tau_j - s, m_j^{(0)})^{-1} = \bigoplus_{j=1}^{\widehat{k_0}} \sum_{i=0}^{m_j^{(0)}-1} \frac{-1}{(s-\tau_j)^{i+1}} \big(S_{m_j^{(0)}}\big)^i$$
$$= \bigoplus_{j=1}^{\widehat{k_0}} \sum_{i=1}^{m_j^{(0)}} \frac{-1}{(s-\tau_j)^{i}} \big(S_{m_j^{(0)}}\big)^{i-1}.$$

It follows similarly

$$(-sJ^{(1)} + I)^{-1} = \bigoplus_{j=1}^{\widehat{k_1}} \sum_{i=0}^{m_j^{(1)}-1} s^i \big(S_{m_j^{(1)}}\big)^i.$$

Inserting the last two equations into Eq. (3.6) gives

$$\boldsymbol{e}_1^{\mathsf{T}} T(s)^{-1} \boldsymbol{e}_1 = \sum_{j=1}^{k_0} \sum_{i=1}^{m_j^{(0)}} \frac{1}{(s-\tau_j)^i} \underbrace{(-\boldsymbol{v}^{(j)})^{\mathsf{T}} \big(S_{m_j^{(0)}}\big)^{i-1} \boldsymbol{u}^{(j)}}_{=w_{j,i}}$$
$$+ \sum_{j=1}^{k_1} \sum_{i=0}^{m_j^{(1)}-1} s^i \underbrace{(\boldsymbol{v}^{(k_0+j)})^{\mathsf{T}} \big(S_{m_j^{(1)}}\big)^{i} \boldsymbol{u}^{(k_0+j)}}_{=\sigma_{j,i}}. \qquad \square$$

Theorem 3.14 tells us that whenever we have a two-term CF-matrix for $r(A)$, we also know how to construct the partial fraction expansion Eq. (3.5): We need to construct the Weierstrass canonical form of the pencil $T_m(s)$. Similarly to the Jordan canonical form, this computation can be prone to numerical instability. We do not expect this to be a problem, however: If the continued fraction is a contracted regular C-fraction and has only positive coefficients, then $T_m^{(0)} = I$ and $T_m^{(1)}$ is symmetric, see Example 3.10. The Weierstrass canonical form now degenerates to the eigendecomposition of a real symmetric matrix, which is a perfectly well-conditioned problem. More generally, if one of $T_m^{(0)}$ or $T_m^{(1)}$ is Hermitian positive definite and the other Hermitian, then we can invoke Theorem 2.86 and the Weierstrass canonical form is again well-conditioned. Note that for two-term CF-matrices obtained by the Euclidean algorithm (Example 3.11), the pencil can again be made symmetric. By not including the polynomial part in the CF-matrix, we can also force $T_m^{(1)}$ to be non-singular by Theorem 3.14.

We are not aware of any relevant rational function that yields a two-term pencil but whose partial fraction expansion is ill-conditioned. Because of this, we assume in the following that the partial fraction expansion is available. Thus, we compare our CF-matrix approach for $r(A)\boldsymbol{b}$ with Eq. (3.5).

**Remark 3.15.** In [CF, Example 5.1], we argued that one can sometimes avoid complex arithmetic with the CF-matrix. The argument went like this: If $T^{(0)}$, $T^{(1)}$, $A$ and $\boldsymbol{b}$ are real, then we can solve $T(A)^{-1}(\boldsymbol{e}_1 \otimes \boldsymbol{b})$ using only real arithmetic. However, specp$(T(s))$ is not necessarily a subset of the real numbers. If it is not, then $A - \tau_j I$ is a complex matrix for some $\tau_j \in$ specp$(T(s))$ and we have to use the more expensive complex arithmetic.

In the special case of Krylov subspace methods, this argument is moot: As the Krylov subspace is shift-invariant, one can still construct it without relying on complex arithmetic by just using the matrix-vector products with $A$ instead of $A - \tau_j I$. The cost-critical part of the algorithm then uses only real numbers. Another counterpoint is valid for all methods: Under our assumptions, the eigenvalues $\tau_j$ come in complex conjugate pairs since the pencil is real, i.e.,

$$(T^{(0)} - \tau_j T^{(1)})\boldsymbol{v} = 0 \implies (T^{(0)} - \overline{\tau_j} T^{(1)})\overline{\boldsymbol{v}} = 0.$$

If we have solved

$$\boldsymbol{x} = (A - \tau_j I)^{-1}\boldsymbol{b},$$

it is not necessary to solve

$$\boldsymbol{y} = (A - \overline{\tau_j} I)^{-1}\boldsymbol{b},$$

for the solution $\boldsymbol{y}$ is just the complex conjugate of $\boldsymbol{x}$, i.e., $\boldsymbol{y} = \overline{\boldsymbol{x}}$. This way, we obtain the solution of two systems effectively for the price of one. It is thus questionable whether the CF-matrix approach would yield a significant speedup merely by avoiding complex arithmetic.

### 3.2.2. Generalized Sylvester equation

The linear system $T(A)\boldsymbol{x} = \boldsymbol{e}_1 \otimes \boldsymbol{b}$ is equivalent to a matrix equation. We state the connection and shortly discuss whether this insight offers a solution method.

We define the *vectorization* $\operatorname{vec}(X)$ of a matrix $X = [\boldsymbol{x}_1 \dots \boldsymbol{x}_m] \in \mathbb{C}^{n\times m}$ as stacking the columns $\boldsymbol{x}_i \in \mathbb{C}^n$ atop of each other, i.e.,

$$\operatorname{vec}(X) = \begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_m \end{bmatrix} \in \mathbb{C}^{nm}.$$

The vectorization and the Kronecker product have the following well-known connection:

**Lemma 3.16** ([57, Corollary 4.3.1])**.** *Let $A \in \mathbb{C}^{p\times n}$, $B \in \mathbb{C}^{m\times q}$ and $X \in \mathbb{C}^{n\times m}$. Then*

$$\operatorname{vec}(AXB) = (B^{\mathsf{T}} \otimes A)\operatorname{vec}(X).$$

With this, we can rewrite the linear system involving the CF-matrix:

**Lemma 3.17** (cf. [CF, Corollary 4.3])**.** *Let*

$$T(A) = \sum_{j=0}^{\ell} T^{(j)} \otimes A^j.$$

*Then the solution* $\boldsymbol{x}$ *of the linear system*

$$T(A)\boldsymbol{x} = \boldsymbol{e}_1 \otimes \boldsymbol{b}$$

*is the vectorization* $\boldsymbol{x} = \operatorname{vec}(X)$ *of the matrix* $X$ *fulfilling*

$$\sum_{j=0}^{\ell} A^j X (T^{(j)})^{\mathsf{T}} = \boldsymbol{b}\boldsymbol{e}_1^{\mathsf{T}}.$$

If $T(A) = T^{(0)} \otimes I - T^{(1)} \otimes A$ and $T^{(1)}$ is non-singular, we obtain the matrix equation

$$X(T^{(0)})^{\mathsf{T}}(T^{(1)})^{-\mathsf{T}} - AX = \boldsymbol{b}\boldsymbol{e}_1^{\mathsf{T}}(T^{(1)})^{-\mathsf{T}}. \tag{3.7}$$

Matrix equations of the form

$$XB - AX = C, \tag{3.8}$$

where $A$, $B$, $C$ and $X$ are matrices with compatible sizes and $X$ is unknown, are called *Sylvester equations*. We see that Eq. (3.7) is a Sylvester equation. For singular $T^{(1)}$, we have

$$X(T^{(0)})^{\mathsf{T}} - AX(T^{(1)})^{\mathsf{T}} = \boldsymbol{b}\boldsymbol{e}_1^{\mathsf{T}},$$
$$XB^{(0)} - AXB^{(1)} = C,$$

which can be called a *generalized Sylvester equation*.

One might now wonder if solution methods for (generalized) Sylvester equations offer any benefit for our case. Algorithmic approaches for such matrix equations are reviewed in [83]. The two main ideas presented there are the following:

- Compute the eigendecomposition of the matrix $B^{\mathsf{T}} = (T^{(1)})^{-1}T^{(0)}$. This leads to shifted linear systems with the matrix $A$, i.e., to the partial fraction expansion according to Theorem 3.14. The Schur decomposition[2] is mentioned as an alternative but involves more shifted linear systems than the eigendecomposition. Since we assume that the partial fraction expansion is available, we do not gain anything from this approach.

- Project the equation onto a subspace. One defines the residual

$$R_k \coloneqq X_k B^{(0)} - AX_k B^{(1)} - C$$

[2]The Schur decomposition yields a triangular matrix $S = UAU^{\mathsf{H}}$ with unitary $U$.

and demands that $V_k^{\mathsf{H}} R_k = 0$, where range$(V_k)$ is some subspace. One obtains the smaller generalized Sylvester equation

$$Y_k B^{(0)} - (V_k^{\mathsf{H}} A V_k) Y_k B^{(1)} = V_k^{\mathsf{H}} C$$

by choosing $X_k = V_k Y_k$, hoping that $X_k \approx X$. For the subspace, it is suggested in [83] to use the block Krylov subspace

$$\mathcal{K}_k^{\square}(A, C) = \text{range}([C, AC, \ldots, A^{k-1}C]).$$

Note that in our case, range$(C) = \text{span}(\boldsymbol{b})$, so

$$\mathcal{K}_k^{\square}(A, C) = \text{range}([\boldsymbol{b}, A\boldsymbol{b}, \ldots, A^{k-1}\boldsymbol{b}]) = \mathcal{K}_k(A, \boldsymbol{b}),$$

i.e., the block Krylov subspace for the Sylvester equation is just the regular Krylov subspace for $A$ and $\boldsymbol{b}$. Moreover, this approach is the same as the Galerkin approach (cf. Lemma 2.79) applied to the linear system $T(A)\boldsymbol{x} = \boldsymbol{e}_1 \otimes \boldsymbol{b}$, i.e.,

$$(I \otimes V_k^{\mathsf{H}}) T(A)(I \otimes V_k)\boldsymbol{y}_k = T(V_k^{\mathsf{H}} A V_k)\boldsymbol{y}_k = \boldsymbol{e}_1 \otimes V_k^{\mathsf{H}}\boldsymbol{b}, \quad \boldsymbol{y}_k = \text{vec}(Y_k).$$

We discuss this in the next subsection.

In essence, we can fall back to the familiar case of a linear system without losing any information or benefit.

### 3.2.3. Krylov subspace methods

A straightforward idea to solve the linear system involving the CF-matrix is to use a subspace projection. As $T_m(A)$ is larger than $A$ but has Kronecker structure, it is reasonable to use a structured subspace of the form range$(I \otimes V_k) = \text{range}(V_k)^m$, where $V_k \in \mathbb{C}^{n \times k}$ with $k < n$. The matrix $V_k$ and the subspace range$(V_k)$ can also be used for a subspace projection for the systems of the partial fraction expansion approach. We now show that these two approaches are in fact equivalent.

We assume for now that the subspace has a basis in the form $I \otimes V_k$ with $V_k^{\mathsf{H}} V_k = I$. Then $(I \otimes V_k^{\mathsf{H}}) T_m(A)(I \otimes V_k) = T_m(V_k^{\mathsf{H}} A V_k)$.

**Lemma 3.18.** *Let $V_k \in \mathbb{C}^{n \times k}$ such that $V_k^{\mathsf{H}} V_k = I$ and $\boldsymbol{e}_1^{\mathsf{T}} T_m(s)^{-1} \boldsymbol{e}_1$ is defined on the spectrum of $V_k^{\mathsf{H}} A V_k$. Let $\boldsymbol{b} \in \mathbb{C}^n$ and*

$$\boldsymbol{x}_k = (I \otimes V_k) T_m(V_k^{\mathsf{H}} A V_k)^{-1} (\boldsymbol{e}_1 \otimes V_k^{\mathsf{H}} \boldsymbol{b}).$$

*Then*

$$(\boldsymbol{e}_1^{\mathsf{T}} \otimes I)\boldsymbol{x}_k = \sum_{j=1}^{k_0} \sum_{i=1}^{m_j^{(0)}} w_{j,i} V_k (V_k^{\mathsf{H}} A V_k - \tau_j I)^{-i} V_k^{\mathsf{H}} \boldsymbol{b} + \sum_{j=1}^{k_1} \sum_{i=0}^{m_j^{(1)}-1} \sigma_{j,i} V_k (V_k^{\mathsf{H}} A V_k)^i \boldsymbol{b} \quad (3.9)$$

*with $w_{j,i}$, $\sigma_{j,i}$ as in Theorem 3.14 for $T(s) = T_m(s)$.*

*Proof.* We have

$$(\boldsymbol{e}_1^{\mathsf{T}} \otimes I)(I \otimes V_k) = \boldsymbol{e}_1^{\mathsf{T}} \otimes V_k = V_k(\boldsymbol{e}_1^{\mathsf{T}} \otimes I).$$

This means that

$$\begin{aligned}(\boldsymbol{e}_1^{\mathsf{T}} \otimes I)\boldsymbol{x}_k &= V_k(\boldsymbol{e}_1^{\mathsf{T}} \otimes I)T_m(V_k^{\mathsf{H}}AV_k)^{-1}(\boldsymbol{e}_1 \otimes I)V_k^{\mathsf{H}}\boldsymbol{b} \\ &= V_k\left(\sum_{j=1}^{k_0}\sum_{i=1}^{m_j^{(0)}} w_{j,i}(V_k^{\mathsf{H}}AV_k - \tau_j I)^{-i} + \sum_{j=1}^{k_1}\sum_{i=0}^{m_j^{(1)}-1} \sigma_{j,i}(V_k^{\mathsf{H}}AV_k)^i\right)V_k^{\mathsf{H}}\boldsymbol{b},\end{aligned}$$

where the last equality follows from Theorems 3.3 and 3.14. Expanding the bracket yields the statement. □

Lemma 3.18 tells us what a projection method on the CF-matrix looks like when written via the partial fraction expansion. Consider the opposite direction: We apply a projection method to the shifted linear systems and the polynomial part of the partial fraction expansion of $r(A)\boldsymbol{b}$; for each system and each power of $A$, we use the same subspace range$(V_k)$. Then we obtain exactly the right-hand side of Eq. (3.9). Thus, Lemma 3.18 says there is no difference between the subspace range$(I_m \otimes V_k) = \text{range}(V_k)^m$ in the CF-matrix approach and the subspace range$(V_k)$ in the partial fraction expansion approach.

There is also no advantage when considering the number of computational operations. Applying $I \otimes V_k$ to $T_m(A)$ is as expensive as applying $V_k$ to $A$. But solving the projected problem can be more expensive with the CF-matrix. For the sake of simplicity, let us assume that there is no polynomial part ($k_1 = 0$) and no higher power inverses ($m_j^{(0)} = 1$ for $j = 1, \ldots, k_0$). Then we have to solve $k_0 = m$ systems of the form

$$(V_k^{\mathsf{H}}AV_k - \tau_j I)^{-1}V_k^{\mathsf{H}}\boldsymbol{b}$$

for the partial fraction expansion approach. The number of operations for solving one such linear system is $\mathcal{O}(k^\alpha)$ with $\alpha \geq 1$ and $\alpha$ depending on the method. The total cost follows as $\mathcal{O}(mk^\alpha)$. On the other hand, for the CF-matrix approach, we have to solve one system of the form

$$T_m(V_k^{\mathsf{H}}AV_k)^{-1}(\boldsymbol{e}_1 \otimes V_k^{\mathsf{H}}\boldsymbol{b}).$$

Here, the total cost is $\mathcal{O}((mk)^\alpha) = \mathcal{O}(m^\alpha k^\alpha)$. This is the same as $\mathcal{O}(mk^\alpha)$ only in the best case $\alpha = 1$ and worse for $\alpha > 1$.

The above also holds for the special case range$(V_k) = \mathcal{K}_k(A, \boldsymbol{b})$. Of course, the subspace range$(I_m \otimes V_k) = \mathcal{K}_k(A, \boldsymbol{b})^m$ is not the standard Krylov subspace $\mathcal{K}_k(T_m(A), \boldsymbol{e}_1 \otimes \boldsymbol{b})$ for the system $T_m(A)\boldsymbol{x} = \boldsymbol{e}_1 \otimes \boldsymbol{b}$. So let us compare these two.

**Lemma 3.19.** *Let $T(A) = T^{(0)} \otimes I - T^{(1)} \otimes A$ with $T^{(0)}, T^{(1)} \in \mathbb{C}^{m\times m}$ and $m \geq 2$. Let $\kappa$ be the smallest number such that $\mathcal{K}_{\kappa+1}(A, \boldsymbol{b}) = \mathcal{K}_\kappa(A, \boldsymbol{b})$. If $k < m\kappa$, then*

$$\mathcal{K}_k(T(A), \boldsymbol{e}_1 \otimes \boldsymbol{b}) \subsetneq \mathcal{K}_k(A, \boldsymbol{b})^m.$$

*If $k \geq m\kappa$, then*

$$\mathcal{K}_k(T(A), \boldsymbol{e}_1 \otimes \boldsymbol{b}) \subseteq \mathcal{K}_\kappa(A, \boldsymbol{b})^m.$$

*Proof.* We first show that every block of a vector $\boldsymbol{v} \in \mathcal{K}_k(T(A), \boldsymbol{e}_1 \otimes \boldsymbol{b})$ is an element of $\mathcal{K}_k(A, \boldsymbol{b})$. For this, we show by induction that

$$T(A)^i(\boldsymbol{e}_1 \otimes \boldsymbol{b}) = \sum_{l=0}^{i} \boldsymbol{v}_l^{(i)} \otimes A^l \boldsymbol{b}$$

for some vectors $\boldsymbol{v}_l^{(i)} \in \mathbb{C}^m$ for every $i \geq 0$. The case $i = 0$ is trivial. Assume now that the above equation holds for some $i$. Then we have

$$\begin{aligned}
T(A)^{i+1}(\boldsymbol{e}_1 \otimes \boldsymbol{b}) &= T(A)\left(\sum_{l=0}^{i} \boldsymbol{v}_l^{(i)} \otimes A^l \boldsymbol{b}\right) \\
&= (T^{(0)} \otimes I)\left(\sum_{l=0}^{i} \boldsymbol{v}_l^{(i)} \otimes A^l \boldsymbol{b}\right) - (T^{(1)} \otimes A)\left(\sum_{l=0}^{i} \boldsymbol{v}_l^{(i)} \otimes A^l \boldsymbol{b}\right) \\
&= \sum_{l=0}^{i} T^{(0)} \boldsymbol{v}_l^{(i)} \otimes A^l \boldsymbol{b} - \sum_{l=1}^{i+1} T^{(1)} \boldsymbol{v}_{l-1}^{(i)} \otimes A^l \boldsymbol{b} \\
&= T^{(0)} \boldsymbol{v}_0^{(i)} \otimes \boldsymbol{b} + \sum_{l=1}^{i} (T^{(0)} \boldsymbol{v}_l^{(i)} - T^{(1)} \boldsymbol{v}_{l-1}^{(i)}) \otimes A^l \boldsymbol{b} - T^{(1)} \boldsymbol{v}_i^{(i)} \otimes A^{i+1} \boldsymbol{b} \\
&= \sum_{l=0}^{i+1} \boldsymbol{v}_l^{(i+1)} \otimes A^l \boldsymbol{b}
\end{aligned}$$

with

$$\boldsymbol{v}_l^{(i+1)} = \begin{cases} T^{(0)} \boldsymbol{v}_0^{(i)}, & l = 0, \\ T^{(0)} \boldsymbol{v}_l^{(i)} - T^{(1)} \boldsymbol{v}_{l-1}^{(i)}, & l = 1, \ldots, i, \\ -T^{(1)} \boldsymbol{v}_i^{(i)}, & l = i+1. \end{cases}$$

This concludes the induction process.

By applying this knowledge to our vector $\boldsymbol{v} \in \mathcal{K}_k(T(A), \boldsymbol{e}_1 \otimes \boldsymbol{b})$, we can write it similarly as a sum of Kronecker products with $A^i \boldsymbol{b}$. Specifically, there are constants $c^{(i)}$ for $i = 0, \ldots, k-1$ such that

$$\boldsymbol{v} = \sum_{i=0}^{k-1} c^{(i)} T(A)^i (\boldsymbol{e}_1 \otimes \boldsymbol{b}) = \sum_{i=0}^{k-1} \sum_{l=0}^{i} c^{(i)} \boldsymbol{v}_l^{(i)} \otimes A^l \boldsymbol{b} = \sum_{l=0}^{k-1} \sum_{i=l}^{k-1} c^{(i)} \boldsymbol{v}_l^{(i)} \otimes A^l \boldsymbol{b} = \sum_{l=0}^{k-1} \tilde{\boldsymbol{v}}_l \otimes A^l \boldsymbol{b}.$$

We obtain the $j$th block by multiplying with $\boldsymbol{e}_j^\mathsf{T} \otimes I$, which leads to

$$(\boldsymbol{e}_j^\mathsf{T} \otimes I)\boldsymbol{v} = \sum_{l=0}^{k-1} (\boldsymbol{e}_j^\mathsf{T} \tilde{\boldsymbol{v}}_l) A^l \boldsymbol{b} \quad \in \quad \mathcal{K}_k(A, \boldsymbol{b}).$$

As this holds for every block, it follows that $\mathcal{K}_k(T(A), \boldsymbol{e}_1 \otimes \boldsymbol{b}) \subseteq \mathcal{K}_k(A, \boldsymbol{b})^m$.

What is left to prove is the inequality of the sets for the case $k < m\kappa$. We use a dimensional argument:

$$\dim(\mathcal{K}_k(T(A), \boldsymbol{e}_1 \otimes \boldsymbol{b})) \leq k < \min\{mk, m\kappa\} = \dim(\mathcal{K}_k(A, \boldsymbol{b})^m)$$

i.e., the subspace $\mathcal{K}_k(T(A), \boldsymbol{e}_1 \otimes \boldsymbol{b})$ is smaller than $\mathcal{K}_k(A, \boldsymbol{b})^m$. □

Therefore, we expect the subspace $\mathcal{K}_k(A, \boldsymbol{b})^m$ to always yield better approximations than $\mathcal{K}_k(T_m(A), \boldsymbol{e}_1 \otimes \boldsymbol{b})$. Both subspaces need exactly $k-1$ matrix-vector products with $A$ to be constructed and we assume that these products are the dominating cost. Accordingly, we do not expect any computational advantage from using $\mathcal{K}_k(T_m(A), \boldsymbol{e}_1 \otimes \boldsymbol{b})$.

The situation is less clear if the residual is less structured. For example, this is the case for a non-zero initial guess or restarts since $T_m(V_k^{\mathsf{H}} A V_k)^{-1}(\boldsymbol{e}_1 \otimes V_k^{\mathsf{H}} \boldsymbol{b})$ is generally not a Kronecker product. But then for a matrix-vector product of $T_m(A)$ and the residual, we need up to $m$ matrix-vector products of $A$ (corresponding to the $m$ blocks). This is in contrast to the partial fraction expansion approach, where we need only one product with $A$ to increase the Krylov subspace even after restarts.

Still, one might hope that some method converges significantly faster if applied to $T(A)$. In this regard, it is interesting that we can easily connect the spectrum of $T(A)$ with the ones of $A - \tau_j I$ in some cases.

**Lemma 3.20.** *Let $T^{(0)}, T^{(1)} \in \mathbb{C}^{m\times m}$ be such that they are simultaneously triangularizable and $T(s) = T^{(0)} - sT^{(1)}$ is a regular pencil. Let $\tau_j$ for $j = 1, \ldots, k_0 \leq m$ denote the eigenvalues of $T(s)$ (with $k_0$ as in its Weierstrass canonical form). Then there exist $c_j \in \mathbb{C}\backslash\{0\}$ such that*

$$\bigcup_{j=1}^{k_0} c_j \operatorname{spec}(A - \tau_j I) \subseteq \operatorname{spec}(T(A)),$$

*where the set equality holds if $T^{(1)}$ is non-singular.*

*Proof.* Since $T^{(0)}$, $T^{(1)}$ are simultaneously triangularizable, there exists a $U$ such that $UT^{(0)}U^{-1}$ and $UT^{(1)}U^{-1}$ are upper triangular. Let $\phi_j$ and $\psi_j$ for $j = 1, \ldots, m$ denote the diagonal elements of $UT^{(0)}U^{-1}$ and $UT^{(1)}U^{-1}$, respectively. We apply the similarity transformation with $U \otimes I$ to $T(A)$. As the resulting matrix is block upper triangular, we have

$$\operatorname{spec}(T(A)) = \bigcup_{j=1}^{m} \operatorname{spec}(\phi_j I - \psi_j A).$$

Assume that there are $k$ non-zero elements of $\psi_j$. We reorder the ordered pairs $(\phi_j, \psi_j)$ such that these elements appear first. Then we can write

$$\operatorname{spec}(T(A)) = \bigcup_{j=1}^{k} (-\psi_j) \operatorname{spec}(A - \frac{\phi_j}{\psi_j} I) \cup \bigcup_{j=k+1}^{m} \{\phi_j\}.$$

Note that $T^{(1)}$ is non-singular if and only if there is no $j$ such that $\psi_j = 0$, in which case the union on the right is empty.

What remains to show is that $k = k_0$ and $\frac{\phi_j}{\psi_j} = \tau_j$. For this, we apply $U$ to the eigenequation of the pencil $T(s)$,

$$(T^{(0)} - sT^{(1)})\boldsymbol{v} = 0,$$
$$(UT^{(0)}U^{-1} - sUT^{(1)}U^{-1})(U\boldsymbol{v}) = 0.$$

The equation is fulfilled if and only if the matrix on the left side has an eigenvalue 0. But this matrix is upper triangular. Thus, we are looking for those values for $s$ such that the diagonal has a zero element, i.e.,

$$\phi_j - s\psi_j = 0.$$

If $\psi_j = 0$, then the equation is only fulfilled if $\phi_j = 0$. But then we could choose any $s$, which would contradict the hypothesis that the pencil is regular. Consequently, there are $k$ values for $j$ left to investigate. For those, we find $k$ eigenvalues of $T(s)$ as $\tau_j = \frac{\phi_j}{\psi_j}$. There are no other eigenvalues, meaning $k = k_0$, since there are no other possibilities to fulfill the equation and the initial similarity transformation with $U$ did not change the eigenvalues. □

While the spectrum of $T(A)$ by itself does not provide us full insight into the behavior of potential Krylov subspace methods, we can still make two observations: The matrix $T(A)$ can only be positive (or negative) definite if all $A - \tau_j I$ are positive or negative definite. This is not sufficient, however, as the $c_j$ might have conflicting signs or be complex. Moreover, some convergence results for Krylov subspace methods use spectral information. For example, the error bound for CG given in Theorem 2.75 uses the condition number $\kappa(T(A)) = \max_{\lambda_i,\lambda_j \in \mathrm{spec}(T(A))} \lambda_i \lambda_j^{-1}$ (for Hermitian positive definite $T(A)$). As this ratio is not changed by multiplying both eigenvalues with a constant, Lemma 3.20 implies that $\kappa(T(A)) \geq \max_j \kappa(A - \tau_j I)$, so we obtain at best the same error bound for $T(A)$ as for $A - \tau_j I$. Of course, we have assumed here that $T^{(0)}$, $T^{(1)}$ are simultaneously triangularizable. This is not a farfetched case, however: An important class of continued fractions is the class of contracted C-fractions. For those, we can choose $T_m^{(0)} = I$ (Example 3.10), which is simultaneously triangularizable with any $T_m^{(1)}$ by using the Jordan canonical form of the latter.

We conclude that the CF-matrix does not seem to offer benefits when working with Krylov subspace methods.

### 3.2.4. Multigrid methods

In this subsection, we want to investigate if it is possible to construct a multigrid method for the CF-matrix approach that beats a good multigrid method for the partial fraction expansion approach. We explained in Section 2.5.2 that we need two components for a multigrid method: a simple iterative method called the smoother and a subspace

projection called the coarse-grid correction. We know that the best possible convergence rate and the according subspace is characterized by the eigendecomposition of the smoother, see Theorem 2.80. Therefore, we aim to relate the eigendecompositions of the two approaches.

As a reminder, we want to solve the linear system

$$T(A)\boldsymbol{x} = \boldsymbol{e}_1 \otimes \boldsymbol{b}$$

and consider iterative methods of the form

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \mathfrak{I}(T(A))^{-1}\boldsymbol{r}_k$$

for the smoother. The matrix $\mathfrak{I}(T(A))$ needs to be cheap to invert but still a good approximation of $T(A)^{-1}$.

Consider the Kronecker product structure of $T(A) = T^{(0)} \otimes I - T^{(1)} \otimes A$. As the first dimension $m$ is assumed to be much smaller than the second dimension $n$, we expect that changes in the first Kronecker factors would not help much. For example, if we choose

$$\mathfrak{I}(T(A)) = (I \odot T^{(0)}) \otimes I - (I \odot T^{(1)}) \otimes A,$$

then $\mathfrak{I}(T(A))$ is block diagonal and thus easier to invert than the block tridiagonal matrix $T(A)$. Indeed, this is just the block Jacobi method. However, we have to invert the $m$ shifted matrices $T^{(0)}_{i,i}I - T^{(1)}_{i,i}A$ at each iteration. Using the partial fraction expansion, we would also need to solve up to $m$ shifted systems but only once, so the above is usually not a reasonable approach.

Hence, we want to investigate changes in the second Kronecker factor. As the matrix $I$ can be trivially inverted, we assume that $\mathfrak{I}(I) = I$ and only consider modifications $\mathfrak{I}(A)$ of $A$:

$$\mathfrak{I}(T(A)) = T^{(0)} \otimes I - T^{(1)} \otimes \mathfrak{I}(A) = T(\mathfrak{I}(A)).$$

However, if we only modify $A$, then $\mathfrak{I}(A)$ can also be used to construct the smoothers

$$\boldsymbol{y}_{j,k+1} = \boldsymbol{y}_{j,k} + \mathfrak{I}(A - \tau_j I)^{-1}\boldsymbol{r}_{i,k}, \quad j = 1, \dots, k_0,$$

for the shifted systems

$$(A - \tau_j I)\boldsymbol{y}_j = \boldsymbol{b}$$

of the partial fraction expansion (Theorem 3.14): We only need to set $\mathfrak{I}(A - \tau_j I) = \mathfrak{I}(A) - \tau_j I$. This is fairly common: For example, it is fulfilled if $\mathfrak{I}(A - \tau_j I)$ yields the diagonal of $A - \tau_j I$ (resulting in the Jacobi method) or the lower triangular part (resulting in the Gauß-Seidel method). So we ask ourselves: Can the smoother for $T(A)$ behave better than the smoothers for $A - \tau_j I$? We relate the spectra of their error propagators in the following theorem:

**Theorem 3.21** (cf. [CF, Theorem 4.7])**.** *Let $T(s) = T^{(0)} - sT^{(1)}$ be a regular pencil and assume that $T(\mathfrak{I}(A))$ is non-singular. Let*

$$\begin{aligned} E[T(A)] &= I - T(\mathfrak{I}(A))^{-1}T(A), \\ E[A - \tau_j I] &= I - (\mathfrak{I}(A) - \tau_j I)^{-1}(A - \tau_j I), \quad j = 1, \dots, k_0, \end{aligned}$$

*where $\tau_j$ are the eigenvalues of $T(s)$. Let further $S = \{0\}$ if $T^{(1)}$ is singular; otherwise let $S$ be the empty set $S = \varnothing$. Then it holds*

$$\operatorname{spec}(E[T(A)]) = \bigcup_{j=1}^{k_0} \operatorname{spec}(E[A - \tau_j I]) \cup S.$$

*Proof.* Clearly, $\operatorname{spec}(I - M) = 1 - \operatorname{spec}(M)$ for every matrix $M$. It is thus sufficient to prove

$$\operatorname{spec}(T(\mathfrak{I}(A))^{-1}T(A)) = \bigcup_{j=1}^{k_0} \operatorname{spec}((\mathfrak{I}(A) - \tau_j I)^{-1}(A - \tau_j I)) \cup \widetilde{S}$$

with $\widetilde{S} = \{1\}$ if $T^{(1)}$ is singular and $\widetilde{S} = \varnothing$ otherwise. Let

$$UT(s)V = (J^{(0)} - sI) \mathbin{\hat{\oplus}} (I - sJ^{(1)}) \quad \Longleftrightarrow \quad UT^{(0)}V = J^{(0)} \mathbin{\hat{\oplus}} I, \;\; UT^{(1)}V = I \mathbin{\hat{\oplus}} J^{(1)}$$

be the Weierstrass canonical form of $T(s)$. We then have

$$\begin{aligned} (U \otimes I)T(A)(V \otimes I) &= (J^{(0)} \mathbin{\hat{\oplus}} I) \otimes I + (I \mathbin{\hat{\oplus}} J^{(1)}) \otimes A \\ &= (J^{(0)} \otimes I + I \otimes A) \mathbin{\hat{\oplus}} (I \otimes I + J^{(1)} \otimes A) \end{aligned}$$

and similarly for $T(\mathfrak{I}(A))$. We define

$$\begin{aligned} \boldsymbol{y} &= (V^{-1} \otimes I)\boldsymbol{x}, \\ D^{(0)} &= (J^{(0)} \otimes I - I \otimes \mathfrak{I}(A))^{-1}(J^{(0)} \otimes I - I \otimes A), \\ D^{(1)} &= (I \otimes I - J^{(1)} \otimes \mathfrak{I}(A))^{-1}(I \otimes I - J^{(1)} \otimes A). \end{aligned}$$

Note that $T(\mathfrak{I}(A))$ non-singular implies that $J^{(0)} \otimes I + I \otimes \mathfrak{I}(A)$ is non-singular. With the help of these definitions, we apply the following similarity transformation to the eigenequation $T(\mathfrak{I}(A))^{-1}T(A)\boldsymbol{x} = \lambda\boldsymbol{x}$:

$$\begin{aligned} (V^{-1} \otimes I)T(\mathfrak{I}(A))^{-1}(U^{-1} \otimes I)(U \otimes I)T(A)(V \otimes I)\boldsymbol{y} &= \lambda\boldsymbol{y}, \\ (D^{(0)} \mathbin{\hat{\oplus}} D^{(1)})\boldsymbol{y} &= \lambda\boldsymbol{y}. \end{aligned}$$

From this, we obtain

$$\operatorname{spec}(T(\mathfrak{I}(A))^{-1}T(A)) = \operatorname{spec}(D^{(0)}) \cup \operatorname{spec}(D^{(1)}).$$

Note that $J^{(0)}$, $J^{(1)}$ and $I$ are upper triangular, so $(U \otimes I)T(A)(V \otimes I)$ and $(U \otimes I)T(\mathfrak{I}(A))(V \otimes I)$ are block upper triangular. Moreover, the inverse and products of block upper triangular matrices are again block upper triangular. This means that both $D^{(0)}$ and $D^{(1)}$ are block upper triangular. Thus, we can determine their eigenvalues by considering each of their diagonal blocks. The diagonal blocks of $D^{(0)}$ are of the form

$$(\tau_j I - \mathfrak{I}(A))^{-1}(\tau_j I - A) = (\mathfrak{I}(A) - \tau_j I)^{-1}(A - \tau_j I).$$

Hence, it follows that

$$\operatorname{spec}(D^{(0)}) = \bigcup_{j=1}^{k_0} \operatorname{spec}((\mathfrak{I}(A) - \tau_j I)^{-1}(A - \tau_j I)).$$

The set $\widetilde{S}$ is obtained from $D^{(1)}$. We see that its diagonal blocks are $I_n$ (because $J^{(1)}$ has only 0 on its diagonal), so it yields the eigenvalue 1 if it exists. However, we know that

$$UT^{(1)}V = I \hat{\oplus} J^{(1)}$$

and that $U$ and $V$ are non-singular by definition. Consequently, the singular block $J^{(1)}$ and by extension $D^{(1)}$ exist if and only if $T^{(1)}$ is singular. □

**Remark 3.22.** It is assumed in Theorem 3.21 that $T(\mathfrak{I}(A))$ is non-singular. This is equivalent to assuming that $T(s)$ is non-singular for every eigenvalue of $\mathfrak{I}(A)$ by Lemma 3.7. If $T(s) = T_m(s)$ is constructed from a continued fraction $r(s) = g_m(s)$, then this is also equivalent to assuming that $r(s) = \boldsymbol{e}_1^{\mathsf{T}} T_m(s)^{-1} \boldsymbol{e}_1$ is defined on the spectrum of $\mathfrak{I}(A)$.

Let us assume that $T^{(1)}$ is non-singular. Then Theorem 3.21 tells us that the smoother for $T(A)$ behaves exactly as if we combined all smoothers for $A - \tau_j I$ into one method. Ergo, the CF-matrix does not offer any benefit. If $T^{(1)}$ is singular, the error propagator might have in addition the eigenvalue 0. But inspecting the partial fraction expansion tells us that this case only occurs if a polynomial part is included in $T(s)$. Hence, this 0 does not offer any advantage since we could also reduce the size of $T(s)$.

Of course, we know from Theorem 2.80 that not only the eigenvalues but also the eigenvectors are important. For example, $E[T(A)]$ could have a cheap and sparse basis for its eigenvectors with large eigenvalues while $E[A - \tau_j I]$ do not. However, the eigenvectors are also closely connected. To prove this, we first mention a simple lemma.

**Lemma 3.23.** *Let $\boldsymbol{a}, \boldsymbol{c} \in \mathbb{C}^m$ and $\boldsymbol{b}, \boldsymbol{d} \in \mathbb{C}^n$. If*

$$\boldsymbol{a} \otimes \boldsymbol{b} = \boldsymbol{c} \otimes \boldsymbol{d} \neq 0,$$

*then $\boldsymbol{a}$ and $\boldsymbol{c}$ are collinear.*

*Proof.* Reading the equation componentwise yields

$$a_i b_j = c_i d_j, \quad i = 1, \dots, m, \;\; j = 1, \dots, n.$$

Choose $j$ such that $b_j \neq 0$. Then

$$a_i = \frac{d_j}{b_j} c_i.$$

Since this holds for every $i = 1, \dots, m$, we see that $\boldsymbol{a}$ and $\boldsymbol{c}$ are collinear. □

Now we can show a correspondence between eigenvectors of the error propagators.

**Theorem 3.24** (cf. [CF, Corollary 4.8][3])**.** *Let the assumptions of Theorem 3.21 hold. Let $\boldsymbol{v} \neq 0 \in \mathbb{C}^m$, $\boldsymbol{w} \neq 0 \in \mathbb{C}^n$ and $\lambda \neq 0 \in \mathbb{C}$. We have*

$$E[T(A)](\boldsymbol{v} \otimes \boldsymbol{w}) = \lambda \boldsymbol{v} \otimes \boldsymbol{w}$$

*if and only if*

$$T(\tau)\boldsymbol{v} = 0 \;\; \textit{and} \;\; E[A - \tau I]\boldsymbol{w} = \lambda \boldsymbol{w}.$$

*Proof.* We rewrite the two eigenequations as generalized eigenvalue problems: Shifting a matrix does not change its eigenvectors. We use $\mu = 1 - \lambda$ to remove the shifts in the error propagators. We also put $T(\mathfrak{I}(A))$ and $(\mathfrak{I}(A) - \tau I)$ to the right side of the equations, which yields the following equivalent statement:

$$T(A)(\boldsymbol{v} \otimes \boldsymbol{w}) = \mu T(\mathfrak{I}(A))(\boldsymbol{v} \otimes \boldsymbol{w})$$

if and only if

$$T(\tau)\boldsymbol{v} = 0 \;\text{ and }\; (A - \tau I)\boldsymbol{w} = \mu(\mathfrak{I}(A) - \tau I)\boldsymbol{w}.$$

We now prove the forward direction: Starting thus from the upper equation, we have

$$\begin{aligned} T^{(0)}\boldsymbol{v} \otimes \boldsymbol{w} - T^{(1)}\boldsymbol{v} \otimes A\boldsymbol{w} &= \mu T^{(0)}\boldsymbol{v} \otimes \boldsymbol{w} - \mu T^{(1)}\boldsymbol{v} \otimes \mathfrak{I}(A)\boldsymbol{w}, \\ (1 - \mu)T^{(0)}\boldsymbol{v} \otimes \boldsymbol{w} &= T^{(1)}\boldsymbol{v} \otimes (A - \mu\mathfrak{I}(A))\boldsymbol{w}. \end{aligned}$$

Assume for now that $T^{(0)}\boldsymbol{v} \neq 0$. Then we can apply Lemma 3.23. On one hand, this means that $T^{(0)}\boldsymbol{v} = \tau T^{(1)}\boldsymbol{v}$ for some constant $\tau \neq 0$, from which we get

$$T^{(1)}\boldsymbol{v} \neq 0, \quad T(\tau)\boldsymbol{v} = 0.$$

Using this, we obtain

$$\begin{aligned} (1 - \mu)\tau T^{(1)}\boldsymbol{v} \otimes \boldsymbol{w} &= T^{(1)}\boldsymbol{v} \otimes (A - \mu\mathfrak{I}(A))\boldsymbol{w}, \\ (1 - \mu)\tau\boldsymbol{w} &= (A - \mu\mathfrak{I}(A))\boldsymbol{w}. \end{aligned}$$

[3] As far as our analysis concerns, Corollary 4.8 in [CF] is much weaker than the result presented here in Theorem 3.24: There, only the backward direction of the equivalence is considered for only a subset of regular pencils. Note that the result in [CF] does not require $\lambda \neq 0$ but the corresponding eigenvectors are of no interest to us anyway.

We put the terms multiplied by $\mu$ to the left side and the others to the right side. This yields

$$\mu(\mathfrak{I}(A) - \tau I)\boldsymbol{w} = (A - \tau I)\boldsymbol{w}$$

on the other hand and thus proves the statement for $T^{(0)}\boldsymbol{v} \neq 0$.

Let us now consider what happens if $T^{(0)}\boldsymbol{v} = 0$. In this case,

$$0 = T^{(1)}\boldsymbol{v} \otimes (A - \mu\mathfrak{I}(A))\boldsymbol{w}.$$

But $T^{(1)}\boldsymbol{v} = 0$ cannot be true because otherwise $T(s)\boldsymbol{v} = 0$ for every $s$ and we assumed that $T(s)$ is a regular pencil, so we find $(A-\mu\mathfrak{I}(A))\boldsymbol{w} = 0$. This situation is included in the statement with $\tau = 0$. We have thus proven the statement in the forward direction. The other direction is verified by retracing our steps for the case $T^{(0)}\boldsymbol{v} \neq 0$. However, we do not need to treat the case $T^{(0)}\boldsymbol{v} = 0$ differently this time. □

How does that translate to a coarse-grid correction? We know by Theorem 2.80 that the best subspace for the coarse-grid correction is given by the subspace that is spanned by eigenvectors with problematic eigenvalues of $E[T(A)]$. We want to exploit the Kronecker product structure and consequently use a basis of the form $I \otimes W$ for this subspace. But Theorem 3.24 tells us that the subspace with basis $W$ eliminates the same eigenvalues in $E[A - \tau_j I]$. Consequently, any such coarse-grid correction for the CF-matrix approach induces a coarse-grid correction for the partial fraction expansion approach that behaves equally well. We have thus found no reason to use the CF-matrix.

**Remark 3.25.** The smoothers we have considered here fulfill $\mathfrak{I}(T(A)) = T(\mathfrak{I}(A))$, i.e., only the matrix $A$ is modified for the inversion. This is a fairly abstract description of an iterative method, so one might wonder if there is a connection to any of the common smoothers like the Jacobi method or the Gauß-Seidel method. As mentioned in the proof of Lemma 3.7, we can always switch the order in the Kronecker product by permutation. Specifically, there always exists a permutation matrix $P$ such that

$$PT(\mathfrak{I}(A))P^{\mathsf{T}} = P(T^{(0)} \otimes I - T^{(1)} \otimes \mathfrak{I}(A))P^{\mathsf{T}} = I \otimes T^{(0)} - \mathfrak{I}(A) \otimes T^{(1)},$$

see [44, Eq. (1.3.5)], [57, Eq. (4.3.12)]. Now, if we take only the diagonal of $A$, i.e., $\mathfrak{I}(A) = I \odot A$, then $PT(\mathfrak{I}(A))P^{\mathsf{T}}$ is block diagonal and inverting a block diagonal matrix is what happens in the block Jacobi method. The resulting smoother for $T(A)$ can thus be interpreted as the block Jacobi method for a specific—and maybe not intuitive—choice of blocks. Similarly, if we take the lower triangular part of $A$, then $PT(\mathfrak{I}(A))P^{\mathsf{T}}$ is block lower triangular and we obtain the block Gauß-Seidel method for the same choice of blocks.

## 3.3. Numerical Experiments

Our investigations in Section 3.2 offered no explicit method for the CF-matrix to beat the partial fraction expansion. As we focused on exploiting the Kronecker product

structure of $T(A)$ before, we try standard methods instead. Specifically, we want to examine numerically two ideas in this section:

1. When solving the shifted systems of the partial fraction expansion, one might want to exploit the shift-invariance of the Krylov subspace and apply a multi-shift solver. Preconditioning these systems is not a trivial task since the preconditioned Krylov subspaces should ideally also be shift-invariant (see, e.g., [2]). On the other hand, with the CF-matrix we only need to solve one system and thus are free to choose any preconditioner. Not explicitly accounting for the block structure of $T(A)$, standard preconditioners could help our approach be advantageous compared to the partial fraction expansion approach. We investigate this in Sections 3.3.1 and 3.3.2.

2. Algebraic multigrid methods are more or less black-box methods, i.e., the coarse-grid hierarchy is constructed automatically from the entries of the matrix. Such methods may work better with $T(A)$ than with $A - \tau_j I$. We check this in Section 3.3.3.

The following numerical experiments were run on a laptop with Intel® Core™ i7-8650U and 16 GB. They were implemented in the programming language Python (Version 3.8.15) using the package SciPy [91] (Version 1.10.0). Where appropriate, we used functions from its module `sparse`.

### 3.3.1. Preconditioned CG

We want to investigate if CG with a standard preconditioner can give better performance applied to $T(A)$ compared to the matrices $A - \tau_j I$. We choose the regular C-fraction

$$g(s) = b_0 + \operatorname*{\mathrm{K}}_{i=1}^{\infty} \left( \frac{c_i s}{1} \right) \quad \text{with} \quad b_0 = 1, \quad c_1 = \alpha, \quad c_i = \begin{cases} \frac{i-2\alpha}{4(i-1)}, & i = 2, 4, \ldots, \\ \frac{i-1+2\alpha}{4i}, & i = 3, 5, \ldots. \end{cases}$$

This way, we have $g(s) = (s+1)^\alpha$ for $|\arg(s+1)| < \pi$ and its approximants yield corresponding Padé approximations, see [22, Eq. 11.7.1]. We build the symmetric CF-matrix by using the contraction $\tilde{g}_{10}(s)$ of $g_{20}(s)$ (Example 3.10) for $\alpha = -\frac{1}{2}$. This results in

$$T^{(0)} = I_{10}, \quad T^{(1)} = -\frac{1}{4} \begin{bmatrix} 3 & 1 & & \\ 1 & 2 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 2 \end{bmatrix} \in \mathbb{R}^{10 \times 10}.$$

To obtain an approximation for $s^{-1/2}$ instead of for $(s+1)^{-1/2}$, we use $A - I$ as argument, which gives

$$T(A) \coloneqq (T^{(0)} + T^{(1)}) \otimes I - T^{(1)} \otimes A.$$

For $A$, we choose the discrete 2D Laplace operator on a square grid of size $N = 100$ with Dirichlet boundary conditions. That is,

$$A = A_1 \oplus A_1 \in \mathbb{R}^{N^2 \times N^2}, \quad A_1 = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{N \times N}, \tag{3.10}$$

where $\oplus$ denotes the Kronecker sum defined as

$$A \oplus B = A \otimes I_{n_2} + I_{n_1} \otimes B$$

for two matrices $A \in \mathbb{C}^{n_1 \times n_1}$ and $B \in \mathbb{C}^{n_2 \times n_2}$.

As $(T^{(1)})^{-1}(T^{(0)} + T^{(1)}) = (T^{(1)})^{-1} + I$, the shifts $\tau_j$ of the partial fraction expansion can easily be obtained by computing the eigenvalues of $T^{(1)}$. Note that $\tau_j < 0$ and that the matrices $A - \tau_j I$ and $T(A)$ are symmetric positive definite. We apply the CG method to the systems $T(A)^{-1}(\boldsymbol{e}_1 \otimes \boldsymbol{b})$ and $(A - \tau_j I)^{-1}\boldsymbol{b}$ with $\boldsymbol{b} = \boldsymbol{e}_1$ and zero initial guess. We also choose a target accuracy of $10^{-12}$ for the relative residual norm and a maximum number of 200 iterations.

The relative residual norms are plotted in Fig. 3.1 on the left side. Since we shift the spectrum of $A$ away from 0, most of the shifted systems are well conditioned and CG finishes after a few iterations. With the CF-matrix, this performance is not attained by the CG method. Instead, it behaves similarly as with the slowest converging system $A - \tau_j I$.

We now construct a sparse incomplete LU factorization as a preconditioner to speed up the convergence of CG. To be more precise, we use the function `spilu()` of the SciPy package (with a fill factor of 40), which yields an ILUTP preconditioner, see, e.g., [78, Section 10.4.4]. The CG method for $T(A)$ now converges after only 9 iterations. Of course, we had to invest additional CPU time to construct the preconditioner. For a fairer comparison with the partial fraction expansion, we precondition each of the matrices $A - \tau_j I$ in the same way. The resulting residuals are plotted in Fig. 3.1 on the right side. We see that even the slowest case needs only about half as many iterations.

One might argue that constructing several preconditioners for $A - \tau_j I$ could turn out to be more expensive than a single one for $T(A)$. We do not want to give a detailed analysis of the CPU time as that depends heavily on the implementation. However, preconditioning and solving all systems $(A - \tau_j I)^{-1}\boldsymbol{b}$ took about 0.5 s in total, while the same for $T(A)^{-1}(\boldsymbol{e}_1 \otimes \boldsymbol{b})$ took about 32 s. This is a large difference in speed, especially considering that we do not exploit the shift-invariance of the Krylov subspace. From this example, it seems that it is not trivial to find a preconditioner such that working with the CF-matrix becomes more efficient than working with the shifted matrices.

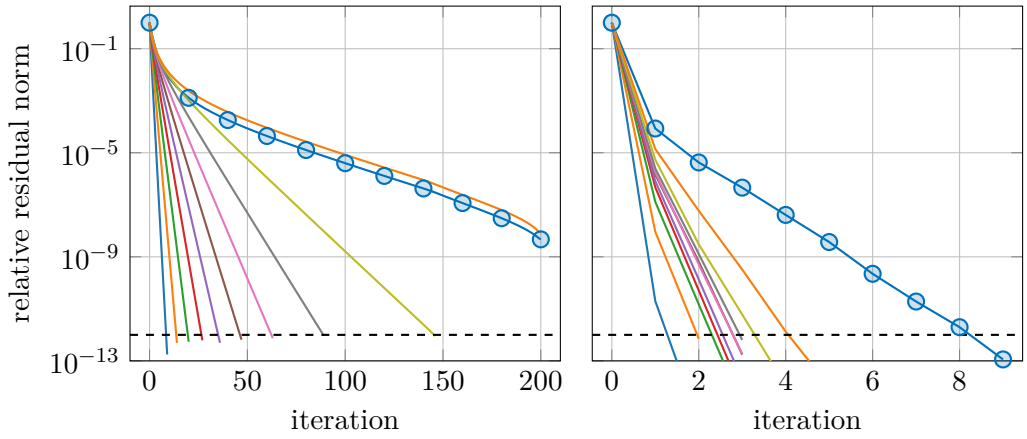


Figure 3.1.: Convergence curves for solving $T(A)^{-1}(\boldsymbol{e}_1 \otimes \boldsymbol{b})$ (marked line, —●—) and $(A - \tau_j I)^{-1}\boldsymbol{b}$ (unmarked lines) with the CG method. The counterclockwise ordering of the unmarked lines corresponds to increasing values of $\tau_j$ and thus increasing condition numbers $\kappa(A - \tau_j I)$. Left: without preconditioner. Right: with ILUTP.

### 3.3.2. Preconditioned GMRES and complex shifts

Perhaps the situation changes if $A$ and $T(A)$ are not symmetric positive definite. We consider the C-fraction

$$g(s) = b_0 + \operatorname*{K}_{i=1}^{\infty} \left(\frac{c_i s}{1}\right) \quad \text{with} \quad b_0 = 1, \quad c_1 = -1, \quad c_i = \begin{cases} (2i-2)^{-1}, & i = 2, 4, \ldots, \\ -(2i)^{-1}, & i = 3, 5, \ldots, \end{cases}$$

which yields $g(s) = \exp(-s)$, see [22, Eq. (11.1.3)]. This time, we have $c_i c_{i+1} < 0$, so we cannot construct a real symmetric $T^{(1)}$ as before. From the contraction $\tilde{g}_{10}(s) = g_{20}(s)$, we construct the CF-matrix as in Example 3.10. We have

$$T^{(0)} = I_{10}, \quad T^{(1)} = (-1) \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & & & \\ -\frac{1}{6} & 0 & \frac{1}{6} & & \\ & -\frac{1}{10} & \ddots & \ddots & \\ & & \ddots & \ddots & \frac{1}{34} \\ & & & -\frac{1}{38} & 0 \end{bmatrix} \in \mathbb{R}^{10\times 10}.$$

We also switch to a non-symmetric (though related) choice for $A$, namely the 2D convection-diffusion operator with first-order upwind discretization. With the step size $h = (N+1)^{-1}$ for a square grid of size $N = 100$ on the unit cube, the direction vector

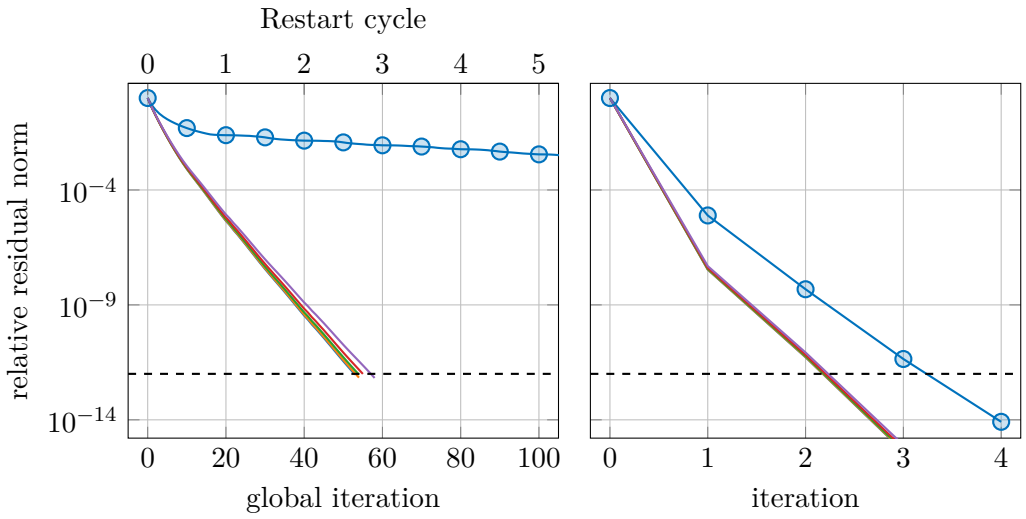


Figure 3.2.: Convergence curves for solving $T(A)^{-1}(\boldsymbol{e}_1 \otimes \boldsymbol{b})$ (marked line, —●—) and $(A - \tau_j I)^{-1}\boldsymbol{b}$ (unmarked lines) with the GMRES method with restart length 20. Left: without preconditioner. Right: with ILUTP, no restarts were necessary.

$[1, -1]$ and a diffusion coefficient of $\epsilon = 10^{-3}$, this implies

$$A = \epsilon h^{-2} A_1 \oplus A_1 + h^{-1} A_2 \oplus A_2^{\mathsf{T}} \in \mathbb{R}^{N^2 \times N^2}, \quad A_2 = \begin{bmatrix} 1 & & & \\ -1 & \ddots & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix} \in \mathbb{R}^{N \times N}.$$

The matrix $A_1$ is defined as in Eq. (3.10). The right-hand side $\boldsymbol{b} = \boldsymbol{e}_1$ and the zero initial guess remain unchanged.

Instead of the CG method, we use the GMRES method with restart length 20 for $T(A)^{-1}(\boldsymbol{e}_1 \otimes \boldsymbol{b})$ and $(A - \tau_j I)^{-1}\boldsymbol{b}$, where

$$T(A) = T^{(0)} \otimes I - T^{(1)} \otimes A, \quad \tau_j \in \operatorname{specp}(T^{(0)} - sT^{(1)}) = \operatorname{spec}((T^{(1)})^{-1}).$$

Note that the shifts $\tau_j$ are all complex numbers that come in complex conjugate pairs. Because of this, we only need to solve five systems of the form $(A - \tau_j I)^{-1}\boldsymbol{b}$, see Remark 3.15. The relative residuals without and with ILUTP are plotted in Fig. 3.2. We see that GMRES with $T(A)$ converges only very slowly, while GMRES with $A - \tau_j I$ finishes in the third restart cycle. With preconditioning, all systems converge within a couple of inner iterations.

For the systems involving $A - \tau_j I$, however, we need complex arithmetic. On the other hand, solving with $T(A)$ is done completely in real arithmetic. As we have discussed in

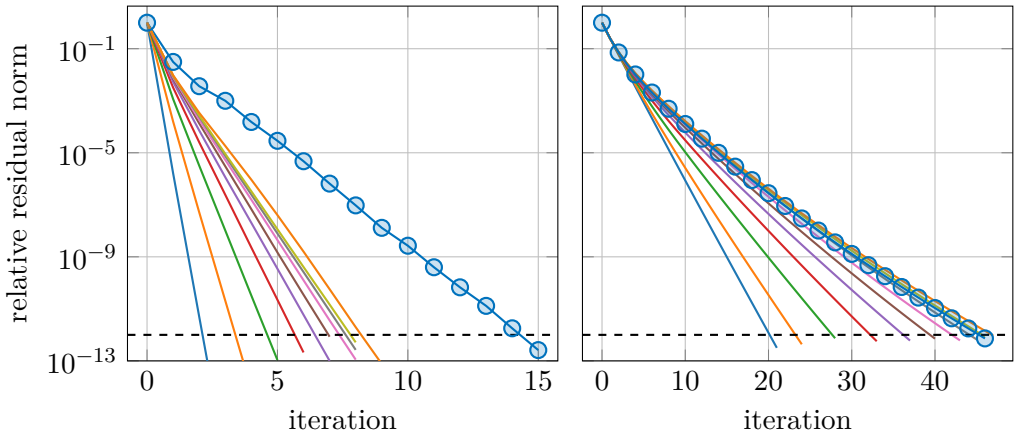


Figure 3.3.: Convergence curves for solving $T(A)^{-1}(\boldsymbol{e}_1 \otimes \boldsymbol{b})$ (marked line, —o—) and $(A - \tau_j I)^{-1}\boldsymbol{b}$ (unmarked lines) with PyAMG. The counterclockwise ordering of the unmarked lines corresponds to increasing values of $\tau_j$ and thus increasing condition numbers $\kappa(A - \tau_j I)$. Left: Black-box approach (`solve()` is used for every system). Right: Structured approach (a coarse-grid hierarchy for $A$ is used for all systems).

Remark 3.15, this could manifest itself by an increased run time for GMRES with $A-\tau_j I$ if the implementation does not exploit the shift-invariance of the Krylov subspace. This is the case for us, so it might be interesting to look at the CPU times: For $T(A)$, constructing the preconditioner and solving with GMRES took about 31 s, while for all 5 systems of the form $A - \tau_j I$, it took about 0.3 s in total. This difference in speed is comparable to before and does not speak in favor of the CF-matrix approach.

### 3.3.3. AMG

Lastly, we choose the function `solve()` from the package PyAMG [9] (Version 4.2.3) for a black-box algebraic multigrid method. We use matrices similar to the ones of Section 3.3.1: The pencil $T(s) \in \mathbb{R}^{10\times 10}$ and the CF-matrix $T(A)$ are constructed in the same way, so implicitly we again try to approximate $A^{-1/2}\boldsymbol{b}$. For $A$, however, we choose the Laplace operator now on a 3D cubic grid of size $N = 30$, i.e.,

$$A = A_1 \oplus A_1 \oplus A_1 \in \mathbb{R}^{N^3\times N^3}$$

with $A_1$ as in Eq. (3.10). The relative residuals for $\boldsymbol{b} = \boldsymbol{e}_1$ are plotted in Fig. 3.3 on the left.

We do not encounter anything new here: While the method converges fast with the CF-matrix, it converges even faster with the shifted systems. If we consider the CPU time, then the black-box method for the CF-matrix took about 23 s whereas for the shifted systems, it finishes after around 14 s. So while still faster than with the CF-

matrix, the partial fraction expansion approach does not seem to work as well with PyAMG.

We can also verify Theorem 3.24 using PyAMG. For this, we choose the Jacobi method as smoother and obtain a coarse-grid hierarchy for the matrix $A$ from PyAMG. (There are several construction methods available. We chose *smoothed aggregation*.) This is used for every shifted matrix $A - \tau_j I$. Lifting the coarse grids of $A$ using the Kronecker product (see Section 3.2.4 and cf. Section 3.2.3) and choosing the block variant of the smoother, we also have a coarse-grid hierarchy for $T(A)$. The relative residuals are plotted in Fig. 3.3 on the right. We see that the multigrid method for $T(A)$ behaves only marginally better than the slowest one for $A - \tau_j I$. We expect that the small difference in the rate of convergence vanishes for a larger number of iterations: After all, the plot implies that a significant portion of the error still comes from faster-converging systems.

# 4. Restarts for Laplace transforms

We discussed in Section 2.4.2 that restarting the Arnoldi approximation to $f(A)\boldsymbol{b}$ is not always a trivial task. Thus far, numerically stable and efficient algorithms are only known for some classes of functions $f$. In this chapter, we show how to make efficient and stable restarts possible for one- and two-sided Laplace transforms and complete Bernstein functions.

Here is our roadmap: In Section 4.1, we develop a new representation of the error function for Laplace transforms and discuss how it can be extended to the other two classes of functions. With this theoretical basis, we explain in Section 4.2 how the restarted Arnoldi method with our error representation can be implemented such that it is indeed stable and efficient in practice. Numerical experiments are presented in Section 4.3. Section 4.4 focuses on error bounds: The new error representation allows us to develop a new a priori error bound, which proves the convergence of our method for some classes of matrices. In addition, we explain how to obtain an a posteriori error bound. We illustrate the practical behavior of these bounds in Section 4.5.

We have already published most of our findings in [L] and Sections 4.1 to 4.3 are based on this. The content of Sections 4.4 and 4.5 has not been published yet, however.

## 4.1. A new representation of the error function

We start with the error representation. As a reminder, if we want to approximate $f(A)\boldsymbol{b}$ using the Arnoldi approximation $\boldsymbol{f}_m$ after $m$ steps, we obtain the error

$$\boldsymbol{\varepsilon}_m = f(A)\boldsymbol{b} - \boldsymbol{f}_m = f(A)\boldsymbol{b} - \|\boldsymbol{b}\|_2 V_m f(H_m)\boldsymbol{e}_1.$$

We now want to find a function $f^{(2)}(s)$ and a vector $\boldsymbol{b}^{(2)}$ such that

$$\boldsymbol{\varepsilon}_m = f^{(2)}(A)\boldsymbol{b}^{(2)}.$$

Once we have found them, we can approximate the error again using the Arnoldi approximation. We show how to find $f^{(2)}(s)$ and $\boldsymbol{b}^{(2)}$ if

$$f(s) = \mathcal{L}\{\hat{f}\}(s) = \int_0^\infty \exp(-ts)\hat{f}(t)\,\mathrm{d}t$$

is the Laplace transform of some function $\hat{f}$. We want $f^{(2)}$ to be a Laplace transform, too, so that we can apply restarts iteratively.

The derivation of the new representation is the subject of Section 4.1.1. We discuss how to translate our result to related classes of functions in Section 4.1.2. This section is based on [L, Sections 3 and 4].

### 4.1.1. Laplace transforms

How do we find a new representation? Looking at the error for a Laplace transform

$$\begin{aligned}\boldsymbol{\varepsilon}_m &= \int_0^\infty \exp(-tA)\boldsymbol{b}\hat{f}(t)\,\mathrm{d}t - \|\boldsymbol{b}\|_2 V_m \int_0^\infty \exp(-tH_m)\boldsymbol{e}_1\hat{f}(t)\,\mathrm{d}t \\ &= \int_0^\infty (\exp(-tA)\boldsymbol{b} - \|\boldsymbol{b}\|_2 V_m \exp(-tH_m)\boldsymbol{e}_1)\hat{f}(t)\,\mathrm{d}t, \\ &= \int_0^\infty \boldsymbol{\varepsilon}_{\exp,m}(t)\hat{f}(t)\,\mathrm{d}t,\end{aligned}$$

we see that we have to integrate the error

$$\boldsymbol{\varepsilon}_{\exp,m}(t) = \exp(-tA)\boldsymbol{b} - \|\boldsymbol{b}\|_2 V_m \exp(-tH_m)\boldsymbol{e}_1 \tag{4.1}$$

for the matrix exponential $\exp(-tA)\boldsymbol{b}$. The idea is to find an integral representation for $\boldsymbol{\varepsilon}_{\exp,m}$ that contains $\exp(-tA)$ and then manipulate the resulting double integral such that we obtain a new matrix Laplace transform. The following representation fulfills our demands and has been known for many years, see, e.g., [27, p. 44] or [92, Proof of Theorem 3.1].

**Lemma 4.1** (cf. [L, Lemma 3.1])**.** *Let $A \in \mathbb{C}^{n\times n}$, $\boldsymbol{b} \in \mathbb{C}^n$. Let $H_m$, $V_m$, $h_{m+1,m}$, $\boldsymbol{v}_{m+1}$ be generated from their Arnoldi process (Algorithm 1). Then*

$$\boldsymbol{\varepsilon}_{\exp,m}(t) = -h_{m+1,m}\|\boldsymbol{b}\|_2 \int_0^t \exp((\tau - t)A)\boldsymbol{v}_{m+1}g(\tau)\,\mathrm{d}\tau,$$

*where $\boldsymbol{\varepsilon}_{\exp,m}(t)$ is defined as in Eq.* (4.1) *and*

$$g(\tau) = \boldsymbol{e}_m^\mathsf{T} \exp(-\tau H_m)\boldsymbol{e}_1$$

*is the $(m, 1)$ entry of $\exp(-\tau H_m)$.*

*Proof.* While the lemma has been used before, it is usually mentioned only en passant. Because of this and for completeness, we give a proof here. We define

$$\boldsymbol{y}(t) = \exp(-tA)\boldsymbol{b}, \quad \boldsymbol{y}_m(t) = \|\boldsymbol{b}\|_2 V_m \exp(-tH_m)\boldsymbol{e}_1$$

so that $\boldsymbol{\varepsilon}_{\exp,m}(t) = \boldsymbol{y}(t) - \boldsymbol{y}_m(t)$. The derivative with respect to $t$ is

$$\boldsymbol{\varepsilon}'_{\exp,m}(t) = \boldsymbol{y}'(t) - \boldsymbol{y}'_m(t) = -A\boldsymbol{y}(t) + \|\boldsymbol{b}\|_2 V_m H_m \exp(-tH_m)\boldsymbol{e}_1.$$

If we insert the Arnoldi relation Eq. (2.3) into the rightmost term, it changes to

$$\begin{aligned}\|\boldsymbol{b}\|_2 V_m H_m \exp(-tH_m)\boldsymbol{e}_1 &= \|\boldsymbol{b}\|_2 (AV_m - h_{m+1,m}\boldsymbol{v}_{m+1}\boldsymbol{e}_m^\mathsf{T}) \exp(-tH_m)\boldsymbol{e}_1 \\ &= A\boldsymbol{y}_m(t) - \|\boldsymbol{b}\|_2 h_{m+1,m}\boldsymbol{v}_{m+1}g(t).\end{aligned}$$

Inserting this back into $\boldsymbol{\varepsilon}'_{\exp,m}(t)$, we see that

$$\boldsymbol{\varepsilon}'_{\exp,m}(t) = -A\boldsymbol{\varepsilon}_{\exp,m}(t) - \|\boldsymbol{b}\|_2 h_{m+1,m}\boldsymbol{v}_{m+1}g(t).$$

We now have an initial-value problem with $\boldsymbol{\varepsilon}_{\exp,m}(0) = \boldsymbol{b} - \boldsymbol{b} = 0$. We solve it using the Laplace transform. Note that $\boldsymbol{\varepsilon}_{\exp,m}(t)$ and $g(t)$ are of exponential type, i.e., we have $\omega\{\boldsymbol{\varepsilon}_{\exp,m}\} < s$ and $\omega\{g\} < s$ for large enough $s$. For such $s$, we can apply the Laplace transform to the above equation, which results in

$$\mathcal{L}\{\boldsymbol{\varepsilon}'_{\exp,m}\}(s) = -A\mathcal{L}\{\boldsymbol{\varepsilon}_{\exp,m}\}(s) - \|\boldsymbol{b}\|_2 h_{m+1,m}\boldsymbol{v}_{m+1}\mathcal{L}\{g\}(s).$$

We know that

$$\mathcal{L}\{\boldsymbol{\varepsilon}'_{\exp,m}\}(s) = s\mathcal{L}\{\boldsymbol{\varepsilon}_{\exp,m}\}(s) - \boldsymbol{\varepsilon}_{\exp,m}(0) = s\mathcal{L}\{\boldsymbol{\varepsilon}_{\exp,m}\}(s)$$

from the Differentiation Theorem for Laplace transforms, see [25, Theorem 9.1]. By combining these two equations, we can write

$$\begin{aligned}(sI + A)\mathcal{L}\{\boldsymbol{\varepsilon}_{\exp,m}\}(s) &= -\|\boldsymbol{b}\|_2 h_{m+1,m}\boldsymbol{v}_{m+1}\mathcal{L}\{g\}(s),\\ \mathcal{L}\{\boldsymbol{\varepsilon}_{\exp,m}\}(s) &= -\|\boldsymbol{b}\|_2 h_{m+1,m}(sI + A)^{-1}\boldsymbol{v}_{m+1}\mathcal{L}\{g\}(s).\end{aligned}$$

As $(sI + A)^{-1} = \mathcal{L}\{\exp(-tA)\}(s)$ (see Example 2.9), we are looking for a function $\boldsymbol{\varepsilon}_{\exp,m}(t)$ whose Laplace transform is the product of two Laplace transforms. According to [25, Theorem 10.1], we obtain the convolution

$$\boldsymbol{\varepsilon}_{\exp,m}(t) = -h_{m+1,m}\|\boldsymbol{b}\|_2 \int_0^t \exp((\tau - t)A)\boldsymbol{v}_{m+1}g(\tau)\,\mathrm{d}\tau,$$

which proves our assertion. □

**Theorem 4.2** (cf. [L, Theorem 3.2])**.** *With the definitions of Lemma 4.1, let $f(s) = \mathcal{L}\{\hat{f}\}(s)$ with measurable $\hat{f}$ be a Laplace transform. Assume that the numerical range $\mathcal{W}(A)$ of $A$ is within the interior of the half-plane of existence, i.e.,*

$$\alpha\{\hat{f}\} < \min_{s\in\mathcal{W}(A)} \mathrm{Re}(s).$$

*Then the error $\boldsymbol{\varepsilon}_m \coloneqq f(A)\boldsymbol{b} - \|\boldsymbol{b}\|_2 V_m f(H_m)\boldsymbol{e}_1$ can be represented as*

$$\boldsymbol{\varepsilon}_m = -h_{m+1,m}\|\boldsymbol{b}\|_2\mathcal{L}\{\hat{f}^{(2)}\}(A)\boldsymbol{v}_{m+1} \quad \textit{with} \quad \hat{f}^{(2)}(t) = \int_0^\infty \hat{f}(t+\tau)g(\tau)\,\mathrm{d}\tau.$$

*Proof.* Starting from the known equality for the error of a Laplace transform

$$\begin{aligned}\boldsymbol{\varepsilon}_m &= \int_0^\infty \exp(-tA)\boldsymbol{b}\hat{f}(t)\,\mathrm{d}t - \|\boldsymbol{b}\|_2 V_m \int_0^\infty \exp(-tH_m)\boldsymbol{e}_1\hat{f}(t)\,\mathrm{d}t\\ &= \int_0^\infty \boldsymbol{\varepsilon}_{\exp,m}(t)\hat{f}(t)\,\mathrm{d}t,\end{aligned}$$

we insert the representation of $\boldsymbol{\varepsilon}_{\exp,m}$ from Lemma 4.1 and obtain

$$\boldsymbol{\varepsilon}_m = -h_{m+1,m}\|\boldsymbol{b}\|_2 \int_0^\infty \int_0^t \exp((\tau - t)A)\boldsymbol{v}_{m+1} g(\tau)\hat{f}(t)\,\mathrm{d}\tau\,\mathrm{d}t.$$

In order to have the exponential of $A$ only in the outer integral, we use the transformation $z = t - \tau$ for the inner integral so that

$$\begin{aligned}\boldsymbol{\varepsilon}_m &= -h_{m+1,m}\|\boldsymbol{b}\|_2 \int_0^\infty \int_0^t \exp(-zA)\boldsymbol{v}_{m+1} g(t - z)\hat{f}(t)\,\mathrm{d}z\,\mathrm{d}t \\ &= -h_{m+1,m}\|\boldsymbol{b}\|_2 \int_0^\infty \int_0^\infty \exp(-zA)\boldsymbol{v}_{m+1} g(t - z)\hat{f}(t)\chi_E(t,z)\,\mathrm{d}z\,\mathrm{d}t. \end{aligned} \tag{4.2}$$

Here, we used the characteristic function

$$\chi_E(t,z) = \begin{cases} 1, & (t,z) \in E, \\ 0, & (t,z) \notin E, \end{cases} \quad \text{with } E = \{(t,z) \in \mathbb{R}^2 : t > z > 0\}.$$

We show in a separate proof in Corollary 4.6 that we can interchange the order of integration. Hence

$$\begin{aligned}\boldsymbol{\varepsilon}_m &= -h_{m+1,m}\|\boldsymbol{b}\|_2 \int_0^\infty \exp(-zA)\boldsymbol{v}_{m+1} \int_0^\infty g(t - z)\hat{f}(t)\chi_E(t,z)\,\mathrm{d}t\,\mathrm{d}z \\ &= -h_{m+1,m}\|\boldsymbol{b}\|_2 \int_0^\infty \exp(-zA)\boldsymbol{v}_{m+1} \int_z^\infty g(t - z)\hat{f}(t)\,\mathrm{d}t\,\mathrm{d}z. \end{aligned}$$

If we transform back, i.e., use $\tau = t - z$ for the inner integral, we have

$$\int_z^\infty g(t - z)\hat{f}(t)\,\mathrm{d}t = \int_0^\infty g(\tau)\hat{f}(z + \tau)\,\mathrm{d}\tau =: \hat{f}^{(2)}(z),$$

which means we proved our assertion. □

What remains to show is that we can indeed interchange the order of integration. We use the following variant of Fubini's Theorem.

**Theorem 4.3** (adapted from Theorem 8.12 in [77])**.** *Let $f(t,z)$ be a measurable function and*

$$\int_X \int_Y |f(t,z)|\,\mathrm{d}z\,\mathrm{d}t < \infty$$

*with $X, Y \subseteq \mathbb{R}$. Then*

$$\int_X \int_Y f(t,z)\,\mathrm{d}z\,\mathrm{d}t = \int_Y \int_X f(t,z)\,\mathrm{d}t\,\mathrm{d}z.$$

We want to apply Theorem 4.3 to Eq. (4.2) and so need to check the two assumptions. We do so in the following two lemmas:

**Lemma 4.4.** *The integrand*

$$\exp(-zA)\boldsymbol{v}_{m+1}g(t-z)\hat{f}(t)\chi_E(t,z)$$

*in Eq.* (4.2) *is measurable.*

*Proof.* Note that the product of measurable functions is also measurable, see [77, 1.9(c)]. Consequently, it suffices to check each factor. The functions $\exp(-zA)$ and $g(t-z)$ are continuous and thus Lebesgue measurable. The function $\hat{f}$ is measurable as otherwise its Laplace transform would not be defined. The characteristic function $\chi_E$ is measurable if $E = \{(t,z) \in \mathbb{R}^2 : t > z > 0\}$ is a measurable set, see [77, 1.9(d)]. To prove this, consider a point $p = (p_x, p_y) \in E$ which implies $p_x > p_y > 0$. Since the rational numbers are dense in $\mathbb{R}$, there is always a rational number $q$ such that $2q > p_x > q > p_y$.[1] The open rectangle $R_q = \{(x,y) \in \mathbb{R}^2 : 2q > x > q > y > 0\}$ is of course measurable (see, e.g., the definition of product $\sigma$-algebras [77, Definition 8.1]). In addition, it contains $p$ and is a subset of $E$. As $p$ has been chosen arbitrarily, $E$ is the union of all such rectangles. But there are only countably many rational numbers $q$ and thus only countably many rectangles $R_q$. We conclude that $E$ is a countable union of measurable sets and thus measurable itself. □

**Lemma 4.5** (cf. [L, Appendix A])**.** *The double integral in Eq.* (4.2) *is finite if the integrand is replaced by its absolute value, i.e.,*

$$\boldsymbol{v} \coloneqq \int_0^\infty \int_0^\infty |\exp(-zA)\boldsymbol{v}_{m+1}g(t-z)\hat{f}(t)\chi_E(t,z)|\,\mathrm{d}z\,\mathrm{d}t < \infty.$$

*Proof.* The integrand and the double integral are vectors, so we need to check every entry individually. To avoid this, we note that $|x_i| \le \|\boldsymbol{x}\|_2$ for every entry $x_i$ of any vector $\boldsymbol{x}$. Applying this and the triangle inequality to the integrand, we have

$$\begin{aligned} v_i &\le \int_0^\infty \int_0^\infty \|\exp(-zA)\boldsymbol{v}_{m+1}\|_2 |g(t-z)|\,|\hat{f}(t)|\,\chi_E(t,z)\,\mathrm{d}z\,\mathrm{d}t \\ &\le \int_0^\infty \int_0^t \|\exp(-zA)\|_2\,|g(t-z)|\,|\hat{f}(t)|\,\mathrm{d}z\,\mathrm{d}t. \end{aligned} \tag{4.3}$$

Similarly, it is hard not to show that

$$\begin{aligned} |g(t)| = |\boldsymbol{e}_m^{\mathsf{T}} \exp(-tH_m)\boldsymbol{e}_1| \le \|\exp(-tH_m)\boldsymbol{e}_1\|_2 \le \|\exp(-tH_m)\|_2, \\ |g(t-z)| \le \|\exp(-(t-z)H_m)\|_2. \end{aligned}$$

The norm of any matrix function can be bounded by

$$\|f(A)\|_2 \le c \sup_{s \in \mathcal{W}(A)} |f(s)|$$

[1] If $2q$ is not large enough, we can find a new rational number $\tilde{q}$ such that $p_x > \tilde{q} > q$ and repeat in this manner until $\tilde{q}$ is close enough to $p_x$.

with $c = 1 + \sqrt{2}$, see [21, Eq. (3), Theorem 3.1]. Applying this to the exponential, we get

$$\|\exp(-zA)\|_2 \leq c \sup_{s \in \mathcal{W}(A)} |\exp(-zs)| = c \max_{s \in \mathcal{W}(A)} \exp(-z \operatorname{Re}(s)) = c \exp(-z\nu)$$

with $\nu = \min_{s \in \mathcal{W}(A)} \operatorname{Re}(s)$. (Note that we need the minimum because $z \geq 0$.) By exploiting that $\mathcal{W}(H_m) \subseteq \mathcal{W}(A)$ and $t - z \geq 0$, we similarly deduce

$$\|\exp(-(t-z)H_m)\|_2 \leq c \exp(-(t-z)\nu).$$

Inserting both bounds into Eq. (4.3) results in

$$\begin{aligned} v_i &\leq c^2 \int_0^\infty \int_0^t \exp(-z\nu) \exp(-(t-z)\nu) |\hat{f}(t)| \, \mathrm{d}z \, \mathrm{d}t \\ &= c^2 \int_0^\infty \exp(-t\nu) |\hat{f}(t)| \int_0^t \mathrm{d}z \, \mathrm{d}t \\ &= c^2 \mathcal{L}\{t|\hat{f}(t)|\}(\nu). \end{aligned}$$

Now, the question is whether $\mathcal{L}\{t|\hat{f}(t)|\}(s)$ exists for $s = \nu$. Theorem 2.8 tells us that $\mathcal{L}\{t|\hat{f}(t)|\}(s)$ is the derivative of $-\mathcal{L}\{|\hat{f}|\}(s)$ and it exists if $\operatorname{Re}(s) > \alpha\{|\hat{f}|\} = \alpha\{\hat{f}\}$. By hypothesis, we have $\alpha\{\hat{f}\} < \nu = \min_{s \in \mathcal{W}(A)} \operatorname{Re}(s)$, so $\mathcal{L}\{t|\hat{f}(t)|\}(\nu)$ exists. □

**Corollary 4.6.** *The order of integration in Eq.* (4.2) *can be interchanged.*

*Proof.* By Lemmas 4.4 and 4.5, we can apply Theorem 4.3. □

Now that we have finished our main proof(s), we want to state the recursive version of Theorem 4.2 in Corollary 4.8. Before that, we have to discuss the region of existence, however. To apply Theorem 4.2 recursively, we need that $\mathcal{W}(A)$ lies again within the new open half-plane $\operatorname{Re}(s) > \alpha\{\hat{f}^{(2)}\}$. It turns out that $\alpha\{\hat{f}^{(2)}\}$ is bounded by $\alpha\{\hat{f}\}$:

**Lemma 4.7.** *Under the assumptions of Theorem 4.2, we have*

$$\alpha\{\hat{f}^{(2)}\} \leq \alpha\{\hat{f}\}.$$

*Proof.* Consider that by definition

$$\mathcal{L}\{\hat{f}^{(2)}\}(s) = \int_0^\infty \int_0^\infty \exp(-zs) g(t-z) \hat{f}(t) \chi_E(t,z) \, \mathrm{d}t \, \mathrm{d}z$$

with $E = \{(t,z) \in \mathbb{R}^2 : t > z > 0\}$. Assume that

$$v \coloneqq \int_0^\infty \int_0^\infty |\exp(-zs) g(t-z) \hat{f}(t) \chi_E(t,z)| \, \mathrm{d}z \, \mathrm{d}t < \infty.$$

Then the double integral

$$\int_0^\infty \int_0^\infty \exp(-zs) g(t-z) \hat{f}(t) \chi_E(t,z) \,\mathrm{d}z \,\mathrm{d}t$$

is finite and it coincides with $\mathcal{L}\{\hat{f}^{(2)}\}(s)$ by Theorem 4.3. (Note that the integrand is measurable, cf. Lemma 4.4.) This implies that $\mathcal{L}\{\hat{f}^{(2)}\}(s)$ exists and thus $\alpha\{\hat{f}^{(2)}\} \leq s$. We now investigate for which $s$ the assumption $v < \infty$ holds.

We proceed similarly to the proof of Lemma 4.5. First, we use the same bound for $|g(t-z)|$ to obtain

$$v \leq c \int_0^\infty |\hat{f}(t)| \exp(-t\nu) \int_0^t \exp(-z(\mathrm{Re}(s) - \nu)) \,\mathrm{d}z \,\mathrm{d}t \tag{4.4}$$

with $\nu = \min_{s \in \mathcal{W}(A)} \mathrm{Re}(s)$ as before. The inner integral is finite for every $s \in \mathbb{C}$. For $\mathrm{Re}(s) = \nu$, we proceed as in the proof of Lemma 4.5, i.e., we have

$$v \leq c \int_0^\infty |\hat{f}(t)| \exp(-t\nu) t \,\mathrm{d}t = c\mathcal{L}\{t|\hat{f}(t)|\}(\nu),$$

which exists by Theorem 2.8. In the case $\mathrm{Re}(s) \neq \nu$, we can evaluate the inner integral in Eq. (4.4) as

$$\int_0^t \exp(-z(\mathrm{Re}(s) - \nu)) \,\mathrm{d}z = \frac{\exp\big(t\nu - t\,\mathrm{Re}(s)\big) - 1}{\nu - \mathrm{Re}(s)}.$$

Inserting this back into Eq. (4.4), we have

$$\begin{aligned} v &\leq \frac{c}{\nu - \mathrm{Re}(s)} \int_0^\infty |\hat{f}(t)| \exp(-t\nu) \big(\exp\big(t\nu - t\,\mathrm{Re}(s)\big) - 1\big) \,\mathrm{d}t \\ &= \frac{c}{\nu - \mathrm{Re}(s)} \left( \int_0^\infty |\hat{f}(t)| \exp(-t\,\mathrm{Re}(s)) \,\mathrm{d}t - \int_0^\infty |\hat{f}(t)| \exp(-t\nu) \,\mathrm{d}t \right) \\ &= \frac{c}{\nu - \mathrm{Re}(s)} \big(\mathcal{L}\{|\hat{f}|\}(\mathrm{Re}(s)) - \mathcal{L}\{|\hat{f}|\}(\nu)\big), \end{aligned}$$

where we assumed that both $\mathcal{L}\{|\hat{f}|\}(\mathrm{Re}(s))$ and $\mathcal{L}\{|\hat{f}|\}(\nu)$ exist for the first equality. $\mathcal{L}\{|\hat{f}|\}(\nu)$ exists by hypothesis (see Theorem 4.2). Thus, $v$ is finite if $\mathcal{L}\{|\hat{f}|\}(\mathrm{Re}(s))$ exists. This requirement is fulfilled for $\mathrm{Re}(s) > \alpha\{\hat{f}\}$. We conclude that $\mathcal{L}\{\hat{f}^{(2)}\}(s)$ exists if $\mathrm{Re}(s) > \alpha\{\hat{f}\}$, which implies $\alpha\{\hat{f}^{(2)}\} \leq \alpha\{\hat{f}\}$. □

**Corollary 4.8** ([L, Corollary 3.3])**.** *Let the assumptions of Theorem 4.2 hold. Let $H_m^{(k)}$, $V_m^{(k)}$, $h_{m+1,m}^{(k)}$, $\boldsymbol{v}_{m+1}^{(k)}$ be generated from the restarted Arnoldi process. Then*

$$\boldsymbol{\varepsilon}_m^{(k)} = \|\boldsymbol{b}\|_2 (-1)^k \Big( \prod_{j=1}^k h_{m+1,m}^{(j)} \Big) \mathcal{L}\{\hat{f}^{(k+1)}\}(A) \boldsymbol{v}_{m+1}^{(k)}, \quad k \geq 1,$$

*where the functions $\hat{f}^{(k+1)}$ are defined by the recursion*

$$\hat{f}^{(1)}(t) = \hat{f}(t),$$
$$\hat{f}^{(k+1)}(t) = \int_0^\infty \hat{f}^{(k)}(t+z) g^{(k)}(z)\,\mathrm{d}z, \quad g^{(k)}(z) = \boldsymbol{e}_m^\mathsf{T} \exp(-zH_m^{(k)})\boldsymbol{e}_1, \quad k \geq 1.$$

### 4.1.2. Related classes of functions

We now turn our attention to other classes of functions for which we can apply Theorem 4.2 or a modification of it.

#### Two-sided Laplace transforms

Two-sided Laplace transforms (Definition 2.17) can be written as the sum of two one-sided Laplace transforms:

$$f(s) = \int_{-\infty}^\infty \exp(-ts)\hat{f}(t)\,\mathrm{d}t = \mathcal{L}\{\hat{f}(t)\}(s) + \mathcal{L}\{\hat{f}(-t)\}(-s).$$

It is straightforward to apply Theorem 4.2 and Corollary 4.8 to both of these separately. In this way, we can implement the restarted Arnoldi method for two-sided Laplace transforms. Most interestingly, the cost remains approximately the same as for a one-sided Laplace transform: We have

$$f(A)\boldsymbol{b} = \mathcal{L}\{\hat{f}(t)\}(A)\boldsymbol{b} + \mathcal{L}\{\hat{f}(-t)\}(-A)\boldsymbol{b},$$

so we need to construct the Krylov subspaces $\mathcal{K}_m(A, \boldsymbol{b})$ and $\mathcal{K}_m(-A, \boldsymbol{b})$. Of course, these two coincide. In essence, although we want to evaluate two (one-sided) Laplace transforms, we need to run the Arnoldi process only once for each restart cycle.

#### Stieltjes functions

An error representation for Stieltjes functions (Definition 2.26) is already known, see Theorem 2.68. Since every Stieltjes function is also a Laplace transform, we can derive this representation in an alternative proof now.

**Lemma 4.9** (cf. Theorem 2.68 and [L, Corollary 3.5])**.** *Under the assumptions of Corollary 4.8, if $f(s)$ is a Stieltjes function with $f(s) = \mathcal{L}\{\mathcal{L}\{\rho\}\}(s)$, then*

$$\boldsymbol{\varepsilon}_m^{(k)} = \|\boldsymbol{b}\|_2 (-1)^k \Big(\prod_{j=1}^k h_{m+1,m}^{(j)}\Big) \int_0^\infty \rho(t) \Big(\prod_{j=1}^k \psi_m^{(j)}(t)\Big) (A + tI)^{-1} \boldsymbol{v}_{m+1}^{(k)}\,\mathrm{d}t,$$

*with $\psi_m^{(j)}(t) = \boldsymbol{e}_m^\mathsf{T}(H_m^{(j)} + tI)^{-1}\boldsymbol{e}_1$.*

*Proof.* Comparing the statement with the representation for $\boldsymbol{\varepsilon}_m^{(k)}$ in Corollary 4.8, we see that the statement reads as

$$\mathcal{L}\{\hat{f}^{(k+1)}\}(A)\boldsymbol{v}_{m+1}^{(k)} = \int_0^\infty \rho(t)\Big(\prod_{j=1}^{k}\psi_m^{(j)}(t)\Big)(A+tI)^{-1}\boldsymbol{v}_{m+1}^{(k)}\,\mathrm{d}t.$$

We simplify this equation by writing the Stieltjes function on the right as a Laplace transform,

$$\mathcal{L}\{\hat{f}^{(k+1)}\}(A)\boldsymbol{v}_{m+1}^{(k)} = \mathcal{L}^2\big\{\rho(t)\prod_{j=1}^{k}\psi_m^{(j)}(t)\big\}(A)\boldsymbol{v}_{m+1}^{(k)},$$

so we only need to show that

$$\hat{f}^{(k+1)}(s) = \mathcal{L}\{\rho(t)\prod_{j=1}^{k}\psi_m^{(j)}(t)\}(s).$$

We prove this by induction for every $k \geq 0$: The case $k = 0$ is trivial for we have $\hat{f}^{(1)} = \hat{f} = \mathcal{L}\{\rho\}$ by hypothesis. Now assume that the above equation holds for $k-1$, i.e.,

$$\hat{f}^{(k)}(s) = \mathcal{L}\{\rho(t)\prod_{j=1}^{k-1}\psi_m^{(j)}(t)\}(s).$$

We can write $\hat{f}^{(k+1)}$ as

$$\hat{f}^{(k+1)}(s) = \int_0^\infty \hat{f}^{(k)}(s+z)g^{(k)}(z)\,\mathrm{d}z = \int_s^\infty \hat{f}^{(k)}(t)g^{(k)}(t-s)\,\mathrm{d}t$$

by applying the transformation $t = s + z$ to its definition (see Corollary 4.8). Since $\hat{f}^{(k)}$ is a Laplace transform by our assumption, we can also write $\hat{f}^{(k+1)}$ as a Laplace transform, see [3, Theorem 2.1] for more details. In particular, we have

$$\hat{f}^{(k+1)}(s) = \mathcal{L}\{\mathcal{L}^{-1}\{\hat{f}^{(k)}\}\mathcal{L}\{g^{(k)}\}\}(s) = \mathcal{L}\{\rho(t)\prod_{j=1}^{k-1}\psi_m^{(j)}(t)\mathcal{L}\{g^{(k)}\}(t)\}.$$

What remains to show is that $\mathcal{L}\{g^{(k)}\} = \psi_m^{(k)}$. This easily follows from Example 2.9 as

$$\begin{aligned}\mathcal{L}\{g^{(k)}\}(t) &= \int_0^\infty \exp(-\tau t)\boldsymbol{e}_m^{\mathsf{T}}\exp(-\tau H_m)\boldsymbol{e}_1\,\mathrm{d}\tau \\ &= \boldsymbol{e}_m^{\mathsf{T}}\mathcal{L}_\tau\{\exp(-\tau t)\}(H_m)\boldsymbol{e}_1 \\ &= \boldsymbol{e}_m^{\mathsf{T}}(H_m^{(k)}+tI)^{-1}\boldsymbol{e}_1 = \psi_m^{(k)}(t).\end{aligned}$$

This concludes the induction and thus our proof. □

### Complete Bernstein functions

Consider a complete Bernstein function

$$f(s) = c_0 + c_1 s + \int_0^\infty (1 - \exp(-ts))\hat{f}(t)\,\mathrm{d}t.$$

Suppose we want to evaluate $f(A)\boldsymbol{b}$. Then the first two terms are given by $c_0\boldsymbol{b}$ and $c_1 A\boldsymbol{b}$ and can be easily evaluated. If we apply the Arnoldi method for the remaining integral, we have

$$\begin{aligned}\boldsymbol{\varepsilon}_m &= \int_0^\infty (I - \exp(-tA))\boldsymbol{b}\hat{f}(t)\,\mathrm{d}t - V_m \int_0^\infty (I - \exp(-tH_m))\boldsymbol{e}_1\hat{f}(t)\,\mathrm{d}t \\ &= \int_0^\infty (\boldsymbol{b} - \exp(-tA)\boldsymbol{b} - (\boldsymbol{b} - V_m \exp(-tH_m)\boldsymbol{e}_1))\hat{f}(t)\,\mathrm{d}t \\ &= -\int_0^\infty (\exp(-tA)\boldsymbol{b} - V_m \exp(-tH_m)\boldsymbol{e}_1)\hat{f}(t)\,\mathrm{d}t.\end{aligned}$$

Provided that $\mathcal{L}\{\hat{f}\}(A)$ is defined, this corresponds to the error for $-\mathcal{L}\{\hat{f}\}(A)\boldsymbol{b}$. We can thus use Theorem 4.2 and Corollary 4.8 for Bernstein functions, as well. We then need to assume that $\mathcal{W}(A)$ lies in the interior of the region of existence of $\mathcal{L}\{\hat{f}\}(t)$. This is quite restrictive. Take $f(s) = \sqrt{s}$ as in Example 2.37. Then we have $\hat{f}(t) = (2\sqrt{\pi})^{-1}t^{-3/2}$. This function is not locally integrable and its Laplace transform accordingly does not exist. Therefore, we mention a variation of Corollary 4.8 explicitly for complete Bernstein functions.

**Lemma 4.10.** *Let*

$$f(s) = \int_0^\infty (1 - \exp(-ts))\hat{f}(t)\,\mathrm{d}t$$

*with*

$$\max(\alpha\{t\hat{f}(t)\}, 0) < \min_{s\in\mathcal{W}(A)} \mathrm{Re}(s).$$

*Then the error for restarted Arnoldi is given by*

$$\boldsymbol{\varepsilon}_m^{(k)} = \|\boldsymbol{b}\|_2(-1)^{k-1}\Big(\prod_{j=1}^{k} h_{m+1,m}^{(j)}\Big)\mathcal{L}\{\hat{f}^{(k+1)}\}(A)\boldsymbol{v}_{m+1}^{(k)}, \quad k \geq 1,$$

*with $\hat{f}^{(j)}$ for $j \geq 1$ as in Corollary 4.8.*

*Proof.* As discussed above, we have the error for the (formal) Laplace transform $-\mathcal{L}\{\hat{f}\}$. For Theorem 4.2, we used the assumption $\alpha\{\hat{f}(t)\} < \min_{s\in\mathcal{W}(A)} \mathrm{Re}(s) =: \nu$ only to prove the existence of $\mathcal{L}\{t|\hat{f}(t)|\}(\nu)$, see the proof of Lemma 4.5. Our new assumption guarantees this, so the case $k = 1$ follows immediately.

To obtain the recursive version ($k > 1$), we show that $\alpha\{\hat{f}^{(2)}\} < \nu$. If this is true, we can just use Corollary 4.8 for all later restarts. Lemma 4.7 does not help us directly

as we have not required anything about $\alpha\{\hat{f}\}$. We follow a similar argument as in its proof, however: We use the bound

$$|g(t-z)| \leq c\exp(-(t-z)\nu) \leq c$$

(note that $(t-z)\nu > 0$) and obtain

$$v \leq c\int_0^\infty |\hat{f}(t)| \int_0^t \exp(-z\operatorname{Re}(s))\,\mathrm{d}z\,\mathrm{d}t$$

instead of Eq. (4.4). Evaluating the inner integral, we have for $s \neq 0$

$$v \leq \frac{c}{\operatorname{Re}(s)}\int_0^\infty |\hat{f}(t)|\big(1-\exp(-t\operatorname{Re}(s))\big)\,\mathrm{d}t,$$

where the integral is just the absolute version of our original Bernstein function $f(s)$. Thus, it is finite for $s > 0$ by definition, which implies that $\alpha\{\hat{f}^{(2)}\} \leq 0$. Since $0 < \nu$, we also have $\alpha\{\hat{f}^{(2)}\} < \nu$ and thus can apply Corollary 4.8. □

**Remark 4.11.** The requirement $0 < \min_{s\in\mathcal{W}(A)} \operatorname{Re}(s)$ is not a large restriction as our definition of a Bernstein function assumes $\operatorname{Re}(s) > 0$, anyway.

**Remark 4.12.** We said that we evaluate $c_0\boldsymbol{b} + c_1A\boldsymbol{b}$ directly. This is actually not necessary: Clearly, $c_0\boldsymbol{b} \in \mathcal{K}_1(A,\boldsymbol{b}) \subseteq \mathcal{K}_2(A,\boldsymbol{b})$ and $c_1A\boldsymbol{b} \in \mathcal{K}_2(A,\boldsymbol{b})$. Hence, assuming that $m \geq 2$, we do not change the error $\boldsymbol{\varepsilon}_m$ by including $c_0 + c_1s$ in the definition of $f(s)$ and Lemma 4.10 still holds.

## 4.2. Implementational aspects

With Corollary 4.8, it seems clear how the restarted Arnoldi method can be implemented for a Laplace transform: We just set

$$f^{(k)}(s) = (-1)^{k-1}\Big(\prod_{j=1}^{k-1} h_{m+1,m}^{(j)}\Big)\mathcal{L}\{\hat{f}^{(k)}\}(s)$$

for $k \geq 2$ in Algorithm 3. However, it turns out there are some open questions regarding an actual implementation, especially if we want it to be efficient. In particular, Line 7 in Algorithm 3, i.e., computing

$$\boldsymbol{f}_m^{(k)} = \|\boldsymbol{b}\|_2 V_m^{(k)} f^{(k)}(H_m^{(k)})\boldsymbol{e}_1,$$

is not as straightforward as it seems. We discuss these questions in this section. It is based on [L, Section 5]. Since our implementation was done in MATLAB [68], we reference some of its functions. The resulting algorithm is stated as Algorithm 4.

**Algorithm 4** Restarted Arnoldi method for Laplace transforms, cf. [L, Algorithm 5.1]

**function** RESTARTED_ARNOLDI($\hat{f}^{(1)}$, $A$, $\boldsymbol{b}$, $m$)
  $V_m^{(1)}$, $H_m^{(1)}$, $h_{m+1,m}^{(1)}$, $\boldsymbol{v}_{m+1}^{(1)}$ = ARNOLDI($A$, $\boldsymbol{b}$, $m$)
  Choose a quadrature rule $(t_i, w_i)_{i=1,\dots,n_\mathrm{q}}$ for $\mathcal{L}\{\hat{f}^{(1)}\}(H_m^{(1)})\boldsymbol{e}_1$.
  $\boldsymbol{d}_m^{(1)} = \boldsymbol{f}_m^{(1)} = \|\boldsymbol{b}\|_2 V_m^{(1)} \mathcal{L}\{\hat{f}^{(1)}\}(H_m^{(1)})\boldsymbol{e}_1$
  **for** $k = 2, 3, \dots$ until convergence **do**
    $V_m^{(k)}$, $H_m^{(k)}$, $h_{m+1,m}^{(k)}$, $\boldsymbol{v}_{m+1}^{(k)}$ = ARNOLDI($A$, $\boldsymbol{v}_{m+1}^{(k-1)}$, $m$)
    Choose a quadrature rule $(t_i, w_i)_{i=1,\dots,n_\mathrm{q}}$ for $\mathcal{L}\{\hat{f}^{(k)}\}(H_m^{(k)})\boldsymbol{e}_1$.
    **if** $k = 2$ **then**
      $p^{(k-1)} = \hat{f}^{(k-1)}$
    **else**
      Construct $p^{(k-1)}$ such that $p^{(k-1)} \approx \hat{f}^{(k-1)}$.
    Approximate $(\hat{f}^{(k)}(t_i))_{i=1,\dots,n_\mathrm{q}}$ using $p^{(k-1)}$ as in Eq. (4.6).
    $c = (-1)^{k-1}\big(\prod_{j=1}^{k-1} h_{m+1,m}^{(j)}\big)$
    Compute $\boldsymbol{f}_m^{(k)} = c\|\boldsymbol{b}\|_2 V_m^{(k)} \mathcal{L}\{\hat{f}^{(k)}\}(H_m^{(k)})\boldsymbol{e}_1$ by numerical quadrature.
    $\boldsymbol{d}_m^{(k)} = \boldsymbol{d}_m^{(k-1)} + \boldsymbol{f}_m^{(k)}$
  **return** $\boldsymbol{d}_m^{(k)}$

### 4.2.1. Quadrature

First, note that for

$$f^{(k)}(H_m^{(k)})\boldsymbol{e}_1 = (-1)^{k-1}\Big(\prod_{j=1}^{k-1} h_{m+1,m}^{(j)}\Big)\mathcal{L}\{\hat{f}^{(k)}\}(H_m^{(k)})\boldsymbol{e}_1,$$

we need to evaluate the Laplace transform $\mathcal{L}\{\hat{f}^{(k)}\}$ at a matrix. In general, we do not have a closed-form expression for this. Thus, we need to rely on numerical integration. We use a quadrature rule to approximate $\mathcal{L}\{\hat{f}^{(k)}\}(H_m^{(k)})\boldsymbol{e}_1$, i.e., we choose $n_\mathrm{q}$ weights $w_i > 0$ and pairwise distinct nodes $t_i \geq 0$ for $i = 1, \dots, n_\mathrm{q}$ and approximate

$$\mathcal{L}\{\hat{f}^{(k)}\}(H_m^{(k)})\boldsymbol{e}_1 \approx \sum_{i=1}^{n_\mathrm{q}} w_i \hat{f}^{(k)}(t_i) \exp(-t_i H_m^{(k)})\boldsymbol{e}_1.$$

How do we choose $w_i$ and $t_i$? Any quadrature rule that gives an accurate approximation can be used. We first describe our implementation but also mention other possibilities.

Note that the integral $\mathcal{L}\{\hat{f}^{(k)}\}(H_m^{(k)})\boldsymbol{e}_1$ is vector-valued (or even matrix-valued if we do not include $\boldsymbol{e}_1$ to the integrand and multiply with it only after the integration). For efficiency, we want to treat every entry in the same way. We can reduce the integral to a scalar-valued one by choosing a number $\nu^{(k)}$ (depending on $H_m^{(k)}$) and determining

the quadrature rule for $\mathcal{L}\{\hat{f}^{(k)}\}(\nu^{(k)})$. As the exponential function quickly converges to 0 for large real values, we expect that the major contributions in $\mathcal{L}\{\hat{f}^{(k)}\}(H_m^{(k)})\boldsymbol{e}_1$ come from eigenvalues with small real value. Thus, we choose $\nu^{(k)}$ to be the smallest real part of the eigenvalues of $H_m^{(k)}$,

$$\nu^{(k)} = \min_{\lambda \in \operatorname{spec}(H_m^{(k)})} \operatorname{Re}(\lambda).$$

This immediately raises the question if a new quadrature rule needs to be set up in every cycle. Our experiments suggest, however, that it is sufficient to use the same rule across all cycles. For the sake of notational simplicity, we assume this in Algorithm 4 and in the following, i.e., $\nu^{(k)} = \nu^{(1)} = \nu$.

In our implementation, we resorted to a well-tried general-purpose quadrature rule: We determine $w_i$ and $t_i$ in the same way as the MATLAB function `integral()` does, see [82]: Two transformations are applied. The first one, $x = \sqrt{t}$, is supposed to treat a possible singularity at $t = 0$, the second one, $z = x(1-x)^{-1}$, yields a finite integration interval. Together, we have $z = \sqrt{t}(1-\sqrt{t})^{-1}$ and

$$\mathcal{L}\{\hat{f}^{(k)}\}(\nu) = \int_0^1 \exp(-\nu\phi(z))\hat{f}^{(k)}(\phi(z))\phi'(z)\,\mathrm{d}z, \quad \phi(z) = \frac{z^2}{(1-z)^2}.$$

The interval $[0,1]$ is partitioned into the 10 subintervals $[0,0.1],[0.1,0.2],\ldots,[0.9,1]$. In each of those, a 7-point Gauß rule is combined with a 15-point Kronrod rule to obtain an approximation to both the integral value on the subinterval and its error.[2] If any estimated error is too large (i.e., larger than a target accuracy $\varepsilon_{\mathrm{q}}$), we split the respective subinterval in the middle and recurse on the new subintervals. The quadrature nodes $z_i$ are then transformed back to $t_i = z_i^2(1+z_i)^{-2}$.

**Remark 4.13.** We observed two problems with the above approach:

- If the imaginary part of an eigenvalue $\lambda$ of $H_m^{(k)}$ is large in absolute value, the quadrature rule defined for $\nu \in \mathbb{R}$ does not yield good approximations to $\mathcal{L}\{\hat{f}\}(\lambda)$. However, if the imaginary part is included in $\nu$, the quadrature can fail to converge or only does with a large number $n_{\mathrm{q}}$ of nodes. This is probably because $\exp(-t\nu) = \exp(-t\operatorname{Re}(\nu))\exp(-ti\operatorname{Im}(\nu))$ then contains the rapidly oscillating term $\exp(-ti\operatorname{Im}(\nu))$. It is indeed advised in [82, Section 4.2] to use specialized methods for oscillatory integrals on infinite intervals.

- We often have that $\exp(-t\nu)\hat{f}(t)$ converges to 0 for $t \to \infty$. However, this is not necessary and $\exp(-t\nu)$ or $\hat{f}(t)$ can grow indefinitely. This can lead to numerical problems if the diverging term evaluates numerically to $\infty$. Thus, one has to be careful, e.g., when working with indefinite or even negative-definite matrices.

[2] For more information about Gauß quadrature rules, see, e.g., [45, Chapter 6] and the references therein.

**Remark 4.14.** The quadrature rule in Line 3 in Algorithm 4 is not needed for Line 4 if $A$ is Hermitian. Then we can compute $f(H_m^{(1)})$ directly via the eigendecomposition of $H_m^{(1)}$. We use it, however, even in the Hermitian case for the evaluation of $\hat{f}^{(2)}$, see Section 4.2.2.

In the literature, numerical integration for the *inverse* Laplace transform—which means finding a function $\hat{f}$ such that $\mathcal{L}\{\hat{f}\}(s) \approx f(s)$ for some value of $s$—has received considerably more attention than integration for the Laplace transform itself. Nonetheless, we want to mention some alternatives to our implementation:

**Example 4.15.** An alternative would be the Gauß-Laguerre quadrature rule. This is a reasonable choice for Laplace transforms since it is exact for integrals of the form

$$\int_0^\infty \exp(-z)p(z)\,\mathrm{d}z$$

if $p(z)$ is a polynomial of degree at most $2n_\mathrm{q} - 1$. This form is obtained for a Laplace transform $\mathcal{L}\{\hat{f}^{(k)}\}(s)$ by using the transformation $z = st$, so that

$$\mathcal{L}\{\hat{f}^{(k)}\}(s) = s^{-1}\int_0^\infty \exp(-z)\hat{f}^{(k)}(s^{-1}z)\,\mathrm{d}z.$$

Now consider our matrix Laplace transforms

$$\mathcal{L}\{\hat{f}^{(k)}\}(H_m^{(k)})\boldsymbol{e}_1 = \int_0^\infty \exp(-z)\hat{f}^{(k)}(z(H_m^{(k)})^{-1})\,\mathrm{d}z\,(H_m^{(k)})^{-1}\boldsymbol{e}_1.$$

We cannot transform the quadrature nodes back because $s$ is a matrix. We notice two problems with this approach:

- Working with the inverse of $H_m^{(k)}$ implies that it is non-singular. We might not be able to guarantee this if $0 \in \mathcal{W}(A)$.
- We have to evaluate the function $\hat{f}^{(k)}$ at the matrix $z(H_m^{(k)})^{-1}$. Thus, $\hat{f}^{(k)}$ needs to be defined on the spectrum of $z(H_m^{(k)})^{-1}$. So if $H_m^{(k)}$ is not diagonalizable, derivatives of $\hat{f}^{(k)}$ need to exist. Note that we also cannot rely on existing optimized algorithms like in Section 4.2.3 for $\exp(-tH_m^{(k)})$.

**Example 4.16.** A recent paper [93] considers numerical integration for both the Laplace transform and its inverse. It is suggested there to truncate an expansion of $\hat{f}$ in terms of scaled Laguerre polynomials $L_i$, which can be transformed exactly. To be more precise, one uses

$$\hat{f}(t) \approx \exp(\alpha t)\sum_{i=0}^{n}\exp(-\beta t)a_i L_i(2\beta t) \implies \mathcal{L}\{\hat{f}\}(s) \approx \frac{1-\hat{r}(s)}{2\beta}\sum_{i=0}^{n} a_i \hat{r}(s)^i$$

with $\alpha > \alpha\{\hat{f}\}$, $\beta > 0$ and

$$\hat{r}(s) = \frac{\alpha + \beta - s}{\alpha - \beta - s},$$

see [93, Section 2.1]. This does not correspond to a quadrature rule. Effectively, we would replace $f(s) = \mathcal{L}\{\hat{f}\}(s)$ by a rational approximation. While this could yield good results in our context, too, it would somewhat defeat the purpose of this chapter: If we approximate $f(s) \approx r(s)$, then we can use the methods discussed in Section 2.5 and do not need a new error representation or algorithm.

**Example 4.17.** Another approach is *sinc quadrature*. The underlying idea is to construct an interpolation of the integrand in terms of shifted sinc functions. We give a short summary of [51, Appendix D]. Let

$$p_h(t) = \sum_{k=-\infty}^{\infty} f(kh)\,\mathrm{sinc}\big(\frac{t}{h} - k\big), \quad \mathrm{sinc}(t) = \frac{\sin(\pi t)}{\pi t}, \quad h > 0.$$

Then it is easily verified that $p_h(ih) = f(ih)$ for $i \in \mathbb{Z}$, i.e., the function $p_h(t)$ interpolates $f(t)$ at the points $ih$. As a first step towards a quadrature rule, one interpolates the integrand in this way. This results in

$$\int_{-\infty}^{\infty} f(t)\,\mathrm{d}t \approx \int_{-\infty}^{\infty} p_h(t)\,\mathrm{d}t = h \sum_{k=-\infty}^{\infty} f(kh)$$

by noting that

$$\int_{-\infty}^{\infty} \mathrm{sinc}\big(\frac{t}{h} - k\big)\,\mathrm{d}t = h \int_{-\infty}^{\infty} \mathrm{sinc}(t)\,\mathrm{d}t = h.$$

The resulting approximation can be interpreted as the infinite trapezoidal rule. Next, one truncates the series in both directions, i.e.,

$$\int_{-\infty}^{\infty} f(t)\,\mathrm{d}t \approx h \sum_{k=-N}^{N} f(kh), \quad N \in \mathbb{N}.$$

If $f(t)$ decays fast enough (e.g., exponentially) for $|t| \to \infty$ and is analytic in a horizontal strip $D_d = \{x + iy : x, y \in \mathbb{R}, -d < y < d\}$, $d > 0$, one can derive how to choose $h$ such that the error is bounded by $\mathcal{O}(\exp(-c\sqrt{N}))$, where $c > 0$ is a constant that does not depend on $N$; see, e.g., [51, Theorem D.28]. If we want to apply this to a Laplace transform $\mathcal{L}\{\hat{f}\}(s)$, we first need to transform the integrand such that the integration interval is $(-\infty, \infty)$ instead of $[0, \infty)$. Note that we cannot use

$$\mathcal{L}\{\hat{f}\}(s) = \int_{-\infty}^{\infty} \exp(-ts)\hat{f}(t)\chi_{[0,\infty)}(t)\,\mathrm{d}t$$

since the integrand would not be analytic at $0 \in D_d$. Whether a suitable transformation is available and, if yes, which one yields the best error bound depends on $\hat{f}$, of course.

For our implementation, we opted for a more black-box approach by relying on the MATLAB function `integral`. We note, however, that there are error bounds for some subclasses of Laplace transforms available, see [86, Theorem 6.9.5]. Furthermore, sinc quadrature for the Stieltjes functions $f(s) = s^{-\alpha}$, $\alpha \in (0, 1)$, with an application to matrix functions has recently been described in [20].

### 4.2.2. Breaking the recursion

Now that we are equipped with a quadrature rule, we can evaluate $\mathcal{L}\{\hat{f}^{(k)}\}(H_m^{(k)})\boldsymbol{e}_1$ by evaluating $\hat{f}^{(k)}(t_i)$ and $\exp(-t_i H_m^{(k)})$ for the quadrature nodes $t_i$, $i = 1, \dots, n_\mathrm{q}$. We now investigate the first part. The matrix exponential is then discussed in Section 4.2.3.

From its definition (see Corollary 4.8), it is easy to see that $\hat{f}^{(k)}$ can be interpreted as yet another Laplace transform,

$$\hat{f}^{(k)}(z) = \int_0^\infty \hat{f}^{(k-1)}(z+t)\boldsymbol{e}_m^\mathsf{T} \exp(-tH_m^{(k-1)})\boldsymbol{e}_1 \,\mathrm{d}t = \boldsymbol{e}_m^\mathsf{T}\mathcal{L}_t\{\hat{f}^{(k-1)}(z+t)\}(H_m^{(k-1)})\boldsymbol{e}_1.$$

The quadrature rule that we used in the previous cycle for $\mathcal{L}\{\hat{f}^{(k-1)}\}(H_m^{(k-1)})\boldsymbol{e}_1$ can be applied to $\hat{f}^{(k)}(z)$ as well. This results in

$$\hat{f}^{(k)}(z) \approx \sum_{i=1}^{n_\mathrm{q}} w_i \hat{f}^{(k-1)}(z+t_i)\boldsymbol{e}_m^\mathsf{T} \exp(-t_i H_m^{(k-1)})\boldsymbol{e}_1.$$

But now the question becomes how to evaluate $\hat{f}^{(k-1)}(z+t_i)$. Indeed, expanding the recursion from the definition of $\hat{f}^{(k)}$,

$$\hat{f}^{(k)}(z) = \int_0^\infty \cdots \int_0^\infty \hat{f}(z + \sum_{i=1}^{k-1} \tau_i) \prod_{i=1}^{k-1} g^{(i)}(\tau_i) \,\mathrm{d}\tau_1 \dots \mathrm{d}\tau_{k-1}, \tag{4.5}$$

we see that we have an $(k-1)$-fold integral. We could naively apply a series of quadrature rules. However, that would become more and more expensive with each new restart cycle.[3] This is not acceptable to us as we demand that our restart algorithm is efficient. We propose to use

$$\hat{f}^{(k)}(z) \approx \sum_{i=1}^{n_\mathrm{q}} w_i p^{(k-1)}(z+t_i)\boldsymbol{e}_m^\mathsf{T} \exp(-t_i H_m^{(k-1)})\boldsymbol{e}_1 \tag{4.6}$$

instead, where $p^{(k-1)} \approx \hat{f}^{(k-1)}$ is some approximation to $\hat{f}^{(k-1)}$ if $k > 2$. As long as the approximation is relatively cheap to compute and its error is small enough in $[0, \infty)$, it is not important what kind of approximation we use. For the remainder of this subsection, we describe how we use interpolating splines in our implementation to give an example.

[3] This can become even more expensive in terms of arithmetic cost than evaluating $\boldsymbol{d}_m^{(k)}$ by Lemma 2.65.

### Spline interpolation

In the previous cycle $k-1$, we have already calculated $\hat{f}^{(k-1)}(t_i)$ for the quadrature rule

$$\mathcal{L}\{\hat{f}^{(k-1)}\}(H_m^{(k-1)})\boldsymbol{e}_1 \approx \sum_{i=1}^{n_\mathrm{q}} w_i \hat{f}^{(k-1)}(t_i) \exp(-t_i H_m^{(k-1)})\boldsymbol{e}_1.$$

For efficiency, we want to reuse those values to construct the approximation $p^{(k-1)}$, which suggests taking $p^{(k-1)}$ as an interpolating function. In our implementation, we opted for an interpolating cubic spline.

**Definition 4.18.** A function

$$p(x) = \begin{cases} p_1(x), & x \le x_2, \\ p_2(x), & x_2 \le x \le x_3, \\ \vdots & \vdots \\ p_{n_\mathrm{s}-1}(x), & x_{n_\mathrm{s}-1} \le x, \end{cases}$$

defined piecewise by cubic polynomials $p_j$ and pairwise distinct nodes $x_i$, is called a *(cubic) spline* if it fulfills the smoothness conditions

$$\begin{aligned} p_{j-1}(x_j) &= p_j(x_j), \\ p'_{j-1}(x_j) &= p'_j(x_j), \\ p''_{j-1}(x_j) &= p''_j(x_j) \end{aligned}$$

for $j = 2, \ldots, n_\mathrm{s} - 1$. A spline is said to *interpolate* $f$ if it satisfies in addition $p(x_j) = f(x_j)$ for $j = 1, \ldots, n_\mathrm{s}$.

Interpolating splines as defined above are not yet unique. Two boundary conditions need to be provided such as the common "not-a-knot" boundary conditions, which mean that $p_1 \equiv p_2$ and $p_{n_\mathrm{s}-2} \equiv p_{n_\mathrm{s}-1}$. The interested reader is referred to [16] for a detailed treatment of spline interpolation. We used the MATLAB function `spline()` to construct such interpolating splines.

In our case, we have $n_\mathrm{s} = n_\mathrm{q}$ and $x_j = t_j$ for $j = 1, \ldots, n_\mathrm{s}$. Initial numerical experiments showed, however, that the approximation error obtained this way tended to be too large, which prevented $\boldsymbol{d}_m^{(k)}$ from converging to $f(A)\boldsymbol{b}$ for $k \to \infty$. (The vectors $\boldsymbol{d}_m^{(k)}$ and subsequently $\boldsymbol{f}_m^{(k)}$ are defined as in Algorithms 3 and 4.)

### Spline refinement

To improve the approximation, we chose to adaptively increase the number $n_\mathrm{s}$ of interpolation nodes. Starting with the spline $p_1^{(k-1)}$ from $x_j = t_j$, we accomplish this by adding the points $(x_j + x_{j+1})/2$ to the set of interpolation nodes, which generates the spline $p_2^{(k-1)}$. We repeat this process—obtaining a more and more refined spline

$p_r^{(k-1)}$ this way—until we are satisfied with the approximation error. We check this by comparing the resulting values for $\boldsymbol{f}_m^{(k)} = \|\boldsymbol{b}\|_2 V_m f^{(k)}(H_m^{(k)})\boldsymbol{e}_1$ after each refinement step. That is, if $\boldsymbol{f}_r$ denotes the computed value of $f^{(k)}(H_m^{(k)})\boldsymbol{e}_1$ with spline $p_r^{(k-1)}$, i.e.,

$$\boldsymbol{f}_r = (-1)^{k-1}\Big(\prod_{j=1}^{k-1} h_{m+1,m}^{(j)}\Big) \underbrace{\underbrace{\sum_{i=1}^{n_\mathrm{q}} w_i \exp(-t_i H_m^{(k)})\boldsymbol{e}_1 \underbrace{\sum_{j=1}^{n_\mathrm{q}} w_j p_r^{(k-1)}(t_i+t_j) g^{(k-1)}(t_j)}_{\approx \hat{f}^{(k)}(t_i)}}_{\approx \mathcal{L}\{\hat{f}^{(k)}\}(H_m^{(k)})\boldsymbol{e}_1}}_{\approx f^{(k)}(H_m^{(k)})\boldsymbol{e}_1},$$

we check whether

$$\|\boldsymbol{b}\|_2 \, \|\boldsymbol{f}_r - \boldsymbol{f}_{r-1}\|_2 \le \varepsilon_\mathrm{s} \|\boldsymbol{d}_m^{(k-1)}\|_2$$

for a user-specified tolerance $\varepsilon_\mathrm{s}$.

We can bound the number of refinement steps in the following way: As the above equation illustrates, we only need to evaluate $p^{(k-1)}$ at $n_\mathrm{q}^2$ values right now: $t_i + t_j$. If the above refinement process adds $\mathcal{O}(n_\mathrm{q}^2)$ points to the set of interpolation nodes, it is favorable to stop the refinement process and add the nodes $t_i + t_j$ instead. Then, the spline $p^{(k-1)}$ does not introduce an additional error anymore (at least in this cycle, see Remark 4.20). It is possible to estimate the number of refinement steps needed, so we can sometimes skip directly to using $t_i + t_j$ as interpolation nodes. We do this with the following error bound for splines:

**Lemma 4.19** (cf. [8, Eq. (1.8)], [L, Section 5.2])**.** *Let $f$ be a real function such that its first three derivatives exist and are continuous in $[x_1, x_{n_\mathrm{s}}]$. Let $p(t)$ be a cubic spline that interpolates $f$ in the points $x_j$, $j = 1, \ldots, n_\mathrm{s}$, with "not-a-knot" boundary conditions. Define $\Delta_j = x_{j+1} - x_j$ and $\Delta_{\max} = \max_{i=1,\ldots,n_\mathrm{s}-1} \Delta_i$. If the fourth derivative $f^{[4]}$ of $f$ exists everywhere in $[x_1, x_{n_\mathrm{s}}]$, then*

$$|f(t) - p(t)| \le c\, \Delta_j^2 \Delta_{\max}^2 \, \|f^{[4]}\|_\infty \quad \textit{for } x_j \le t \le x_{j+1},$$

*where $c > 0$ is a constant independent of $f$ and $x_j$ and*

$$\|f^{[4]}\|_\infty = \sup_{s \in [x_1, x_{n_\mathrm{s}}]} |f^{[4]}(s)|.$$

*Proof.* We start from [8, Eq. (1.8)], i.e.,

$$|f(t) - p(t)| \le c\, \Delta_j \Delta_{\max}^2 \, \delta(f^{[3]}, \Delta_j) \quad \text{for } x_j \le t \le x_{j+1}$$

with

$$\delta(f^{[3]}, \Delta_j) = \sup\{|f^{[3]}(t+h) - f^{[3]}(t)| : \; x_1 \le t \le x_{n_\mathrm{s}},\; 0 < h \le \Delta_j\}.$$

As the fourth derivative of $f$ exists everywhere in $[x_1, x_{n_\mathrm{s}}]$, we can write

$$|f^{[3]}(t+h) - f^{[3]}(t)| = \left| \int_t^{t+h} f^{[4]}(s)\,\mathrm{d}s \right| \leq \int_t^{t+h} |f^{[4]}(s)|\,\mathrm{d}s \leq h\|f^{[4]}\|_\infty,$$

see [77, Theorem 7.21] for the equality. The proof concludes by inserting this into $\delta(f^{[3]}, \Delta_j)$:

$$\delta(f^{[3]}, \Delta_j) \leq \sup\{h\|f^{[4]}\|_\infty : \ 0 < h \leq \Delta_j\} = \Delta_j \|f^{[4]}\|_\infty. \qquad \square$$

The above error bound is reduced by a factor of $2^4 = 16$ for each refinement step. For our estimation, we thus assume that the error at each $t_i + t_j$ is reduced by a factor of 16 in each step. The estimated number of necessary refinement steps $r$ follows as

$$r \approx \log_{16} \frac{\|\boldsymbol{b}\|_2 \, \|\boldsymbol{f}_2 - \boldsymbol{f}_1\|_2}{\varepsilon_\mathrm{s} \|\boldsymbol{d}_m^{(k-1)}\|_2}.$$

Of course, the exact error can differ from the bound (if it is indeed applicable) but we have not observed large deviations in our experiments. Note that the goal of this estimation is to decide whether it is cheaper to skip the refinement process and use $t_i + t_j$ as interpolation nodes instead. Consequently, even if the error bound does not reflect the real behavior of the error, the total number of evaluations of $\hat{f}^{(k-1)}$ is still $\mathcal{O}(n_\mathrm{q}^2)$.

**Remark 4.20.** Assume we have constructed the spline $p^{(k-1)}$ for $\hat{f}^{(k-1)}$ in cycle $k$ and move to the next cycle $k+1$. We now want to construct the spline $p^{(k)}$ for $\hat{f}^{(k)}$ so that

$$\hat{f}^{(k+1)}(z) \approx \sum_{i=1}^{n_\mathrm{q}} w_i p^{(k)}(z + t_i) \boldsymbol{e}_m^\mathsf{T} \exp(-t_i H_m^{(k)}) \boldsymbol{e}_1.$$

For refining $p^{(k)}$, we then need additional evaluations of $\hat{f}^{(k)}$. We reuse $p^{(k-1)}$ instead of $\hat{f}^{(k-1)}$ to keep avoiding the recursion. This leads to evaluations of the form $p^{(k-1)}(t_i + t_j + t_l)$ for $i, j, l = 1, \ldots, n_\mathrm{q}$. Thus, even if $p^{(k-1)}(t_i + t_j) = \hat{f}^{(k-1)}(t_i + t_j)$, the use of spline interpolation typically introduces still an error.

### 4.2.3. Matrix exponential function

If we use a quadrature rule as described in Section 4.2.1, we need to evaluate terms of the form $\exp(-t_j H_m^{(k)})\boldsymbol{e}_1$ for several values of $t_j$. We now discuss how this can be implemented efficiently.

For Hermitian $H_m^{(k)}$ (i.e., for Hermitian $A$), we can use its eigendecomposition as mentioned in Section 2.3: We have

$$\exp(-t_j H_m^{(k)})\boldsymbol{e}_1 = Z \exp(-t_j \Lambda) Z^\mathsf{H} \boldsymbol{e}_1$$

for $H_m^{(k)} = Z\Lambda Z^{\mathsf{H}}$ with diagonal $\Lambda$ and unitary $Z$. The only term that depends on $t_j$ is $\exp(-t_j\Lambda)$, which simply resolves to $m$ evaluations of the (scalar) exponential function. If the eigendecomposition involves $\mathcal{O}(m^3)$ operations, then we need $\mathcal{O}(m^3 + n_{\mathrm{q}}m^2)$ operations in total, for we obtain the matrix-vector product $Z(\exp(-t_j\Lambda)Z^{\mathsf{H}}\boldsymbol{e}_1)$ for a cost of $\mathcal{O}(m^2)$ for $n_{\mathrm{q}}$ values of $t_j$.[4]

If $H_m^{(k)}$ is not Hermitian, the eigendecomposition might be ill-conditioned or not exist at all. MATLAB offers the function `expm()`, which computes the matrix exponential for any matrix. The underlying gist of this function is to use the *scaling and squaring* approach, i.e.,

$$\exp(-t_j H_m^{(k)}) = \exp(-2^{-1}t_j H_m^{(k)})^2 = \exp(-2^{-i}t_j H_m^{(k)})^{2^i}.$$

The matrix exponential $M = \exp(-2^{-i}t_j H_m^{(k)})$ is evaluated by a Padé approximation, which yields a good approximation if the eigenvalues of $-2^{-i}t_j H_m^{(k)}$ are close to 0, i.e., if $i$ is large enough. (The number $i$ is chosen based on a norm estimation of the matrix $t_j H_m^{(k)}$.) Then repeated squaring is used to compute $M^{2^i}$, i.e.,

$$M^{2^i} = (M^2)^{2^{i-1}} = (\dots((M^2)^2)^2\dots)^2,$$

which needs $i$ matrix-matrix products. For more details about `expm()`, see [4, 55] or [56, Section 10.3]. Note that one call to `expm()` involves $\mathcal{O}(m^3)$ operations, so we have $\mathcal{O}(n_{\mathrm{q}}m^3)$ operations in total.

As we are not interested in the full matrix $\exp(-t_j H_m^{(k)})$, we can reduce the execution time by employing an algorithm that does not compute $\exp(-t_j H_m^{(k)})$ explicitly and instead directly computes the vector $\exp(-t_j H_m^{(k)})\boldsymbol{e}_1$. We choose the MATLAB function `expmv()`[5], which implements the algorithm presented in [5]. As we did for `expm()`, we only give a short overview of `expmv()`. Here, only matrix-vector products are needed by switching to truncated Taylor series instead of Padé approximations. Thus, one call should involve $\mathcal{O}(m^2)$ operations. Note that there is no analog of repeated squaring for matrix-vector products, so $M^{2^i}\boldsymbol{v}$ involves $2^i$ matrix-vector products, while $M^{2^i}$ only requires $i$ matrix-matrix products. Thus, we can expect `expmv()` to be computationally cheaper than `expm()` only if it determines a small value for $i$. We modified the code of `expmv()` in our implementation accordingly so that for $i > 0$ it switches to `expm()`. In the worst case, all calls to `expmv()` end up using `expm()`, which still leads to $\mathcal{O}(n_{\mathrm{q}}m^3)$ operations. However, if only a small number of calls to `expm()` are involved, we have again $\mathcal{O}(m^3 + n_{\mathrm{q}}m^2)$ operations.

**Remark 4.21.** One can reduce the probability that scaling needs to be involved in `expmv()`: Assuming that the quadrature nodes $t_j$ are sorted in increasing order, one writes

$$\boldsymbol{x}_j \coloneqq \exp(-t_j H_m^{(k)})\boldsymbol{e}_1 = \exp(-(t_j - t_{j-1})H_m^{(k)})\boldsymbol{x}_{j-1},$$

[4] Of course, the vector $Z^{\mathsf{H}}\boldsymbol{e}_1$ can be reused for all values of $t_j$.

[5] available at `https://github.com/higham/expmv`

This means one can reuse the already computed vector $\boldsymbol{x}_{j-1} \coloneqq \exp(-t_{j-1}H_m^{(k)})\boldsymbol{e}_1$. This way, one needs the matrix exponential of $-(t_j - t_{j-1})H_m^{(k)}$ instead of $-t_j H_m^{(k)}$, which has eigenvalues close to 0 if $t_j \approx t_{j-1}$. Note, however, that this propagates any error in $\boldsymbol{x}_{j-1}$ to all later vectors $\boldsymbol{x}_j, \boldsymbol{x}_{j+1}, \ldots, \boldsymbol{x}_{n_\mathrm{q}}$.

### 4.2.4. Modifications for complete Bernstein functions

Before we move on to numerical experiments, we want to mention how Algorithm 4 needs to be modified such that it can be used for complete Bernstein functions instead of Laplace transforms. We assume that $f$ and $\hat{f}$ are given such that

$$f(s) = \int_0^\infty (1 - \exp(-st))\hat{f}(t)\,\mathrm{d}t.$$

Let us first assume that the matrix $A$ is not Hermitian. Then we need the following changes, which we implemented for our numerical experiment in Section 4.3.4.

- In Line 4, the Laplace transform $\mathcal{L}\{\hat{f}\}$ needs to be replaced by the Bernstein function $f$.
- We might replace Line 3 by two quadrature rules: We need one quadrature rule for the complete Bernstein function $f$ to evaluate $f(H_m^{(1)})$ (a Bernstein function, not a Laplace transform). We need another one for the next cycle to evaluate $\hat{f}^{(2)}(z)$, which we interpret as a Laplace transform for fixed $z$, see Section 4.2.2. Of course, nothing prevents us from trying the same quadrature rule for both cases and this is indeed what we do in our implementation.
- The sign of $\hat{f}$ needs to be flipped after Line 4 (note the additional factor $-1$ in Lemma 4.10 compared to Corollary 4.8).

If $A$ is Hermitian, then we can evaluate $f(H_m^{(1)})$ without the use of a quadrature rule by using the eigendecomposition of $H_m^{(1)}$, cf. Remark 4.14. This means we could potentially leave Line 3 unchanged. However, the Laplace transform $\mathcal{L}\{\hat{f}(t)\}$ might not exist, in which case trying an adaptive quadrature rule as we do is a bad idea. We know that $\mathcal{L}_t\{\hat{f}(t+z)\}(\nu)$ exists for almost all $z \in [0, \infty)$ (with $\nu$ being the smallest eigenvalue of $H_m^{(1)}$). Otherwise, the representation of the error function as a Laplace transform (Lemma 4.10) would not be possible. One alternative is thus to use some fixed value for $z$ to determine the quadrature rule.

For Hermitian $A$, there is in fact no need for any distinction between the two algorithms. We could just change Line 4 to evaluate $f$ instead of $\mathcal{L}\{\hat{f}\}$. Then Algorithm 4 can still be used for Laplace transforms. It is, however, also applicable to complete Bernstein functions: We only need to pass $-\hat{f}$ instead of $\hat{f}$. If $\mathcal{L}\{\hat{f}\}$ does not exist, then we can pass $-\hat{f}(t+\epsilon)$. A black-box quadrature rule has less trouble with that function and it is a good approximation if $\hat{f}$ is continuous and $\epsilon$ is small enough. We try this in Section 4.3.5.

## 4.3. Numerical experiments I: Comparison to other methods

We now want to verify whether we can implement the restarted Arnoldi method for Laplace transforms in an efficient and stable way. We also want to compare if our approach yields any benefit compared to the alternative algorithms:

- Even if $f(s)$ is not a Stieltjes function, we might be able to use the algorithm from [40] after some modification. For example, $h(s) = f(s)s^{-1}$ is a Stieltjes function for many complete Bernstein functions $f(s)$, see Theorem 2.36.[6] If $A$ is non-singular, we can write $f(A)\boldsymbol{b} = h(A)(A\boldsymbol{b})$ and need to evaluate a matrix Stieltjes function.[7]

- In the Hermitian case, we do not have to rely on restarts at all since we can use the two-pass Lanczos method, which we described at the end of Section 2.4.1.

We report the results of numerical experiments that are obtained from our implementation. Sections 4.3.1 and 4.3.2 concern Laplace transforms. In Section 4.3.3, we treat a two-sided Laplace transform. Lastly, we apply our algorithm to two complete Bernstein functions in Sections 4.3.4 and 4.3.5.

All experiments are calculated in MATLAB R2021a [68] on a laptop with Intel® Core™ i7-8650U and 16 GB. For the algorithm for Stieltjes functions from [40], we use the MATLAB package `funm_quad`[8] [42].

As target precision `tol` for the relative error norm, we choose $\texttt{tol} = 10^{-7}$. The absolute and relative error tolerance $\varepsilon_{\mathrm{q}}$ for the quadrature rule is set to $\varepsilon_{\mathrm{q}} = 10^{-3} \cdot \texttt{tol} = 10^{-10}$. The same is done for the tolerance $\varepsilon_{\mathrm{s}}$ for the spline refinement from Section 4.2.2, $\varepsilon_{\mathrm{s}} = \varepsilon_{\mathrm{q}} = 10^{-10}$. Unless otherwise noted, we use a restart length of $m = 50$ for Hermitian matrices and $m = 20$ for other matrices. For the two-pass Lanczos method, we check for convergence every $m$ steps to make it comparable—at least to some extent—to the restarted algorithms. We base our discussion on [L, Section 6], where we have already published these experiments (except for those in Section 4.3.5). Accordingly, we reuse the data for generating the plots.

### 4.3.1. Fractional negative power less than $-1$

This subsection is based on [L, Section 6.1]. As a first function, we choose $f(s) = s^{-3/2}$ as a fractional power less than $-1$. By Example 2.9, we know that

$$f(s) = s^{-3/2} = \frac{2}{\sqrt{\pi}} \mathcal{L}\{\sqrt{t}\}(s)$$

[6] As mentioned there, this holds for *all* complete Bernstein functions for a more general definition of Stieltjes functions.

[7] In some sense, this is possible even for singular $A$, see Remark 4.23.

[8] available at `http://www.guettel.com/funm_quad`

with $\alpha\{\sqrt{t}\} = 0$. The function $\hat{f}(t) = \sqrt{t}$ is not a Laplace transform and thus $f$ is not a Stieltjes function. However, we saw in Example 2.30 that

$$h(s) = f(s) \cdot s = s^{-1/2} = \pi^{-1} \int_0^\infty \frac{1}{\sqrt{t}(t+s)} \, \mathrm{d}t$$

is a Stieltjes function. Consequently, we can evaluate $f(A)\boldsymbol{b}$ by first solving $\boldsymbol{x} = A^{-1}\boldsymbol{b}$ using CG[9] or GMRES and then evaluating $h(A)\boldsymbol{x}$ with `funm_quad`. For CG and GMRES, we use the built-in MATLAB functionalities. We choose a smaller target precision of $10^{-2} \cdot \texttt{tol} = 10^{-9}$ for the residual norm in these methods. This ensures that `funm_quad` can achieve a precision of `tol` even with the initial error for $A^{-1}\boldsymbol{b}$ present.

As the Hermitian test matrix, we choose the 3D Laplace operator from Section 3.3.3:

$$A_{\mathrm{L}} = A_1 \oplus A_1 \oplus A_1 \in \mathbb{R}^{N^3 \times N^3}, \quad A_1 = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{N \times N},$$

where $\oplus$ denotes the Kronecker sum. For a non-Hermitian matrix, we use the convection-diffusion operator with first-order upwind discretization from Section 3.3.2, but we extend it to three dimensions. With the step size $h = (N+1)^{-1}$ for a square grid of size $N$ on the unit cube, the direction vector $[1, -1, 1]$ and a diffusion coefficient of $\epsilon = 10^{-3}$, this results in the matrix

$$A_{\mathrm{CD}} = \epsilon h^{-2} A_{\mathrm{L}} + h^{-1} A_2 \oplus A_2^{\mathsf{T}} \oplus A_2 \in \mathbb{R}^{N^3 \times N^3}, \quad A_2 = \begin{bmatrix} 1 & & & \\ -1 & \ddots & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix} \in \mathbb{R}^{N \times N}.$$

We choose

$$\boldsymbol{b} = \boldsymbol{v} \otimes \boldsymbol{v} \otimes \boldsymbol{v} \quad \text{with } \boldsymbol{v} = \sum_{i=1}^{N} \left[ \sin\Big(\frac{\pi i}{N+1}\Big), \ldots, \sin\Big(\frac{N\pi i}{N+1}\Big) \right]^{\mathsf{T}} \in \mathbb{R}^N$$

in the Hermitian case and $\boldsymbol{b} = [1, \ldots, 1]^{\mathsf{T}} \in \mathbb{R}^{N^3}$ in the non-Hermitian case. In our experiments, we vary $N$ from $N = 20$ to $N = 100$ in steps of 10, which leads to matrices of sizes $n = 8000$ to $n = 10^6$. Note that for both $A_{\mathrm{L}}$ and $A_{\mathrm{CD}}$ and for all values of $N \geq 1$, the numerical range lies strictly in the right half plane. This means our new algorithm can be applied.

Figure 4.1 shows how many matrix-vector products each of the three algorithms perform before reaching the target precision. We see that we need significantly fewer

[9] Note that $\alpha\{\hat{f}\} = 0$ implies that $A$ has to be positive definite, anyway.

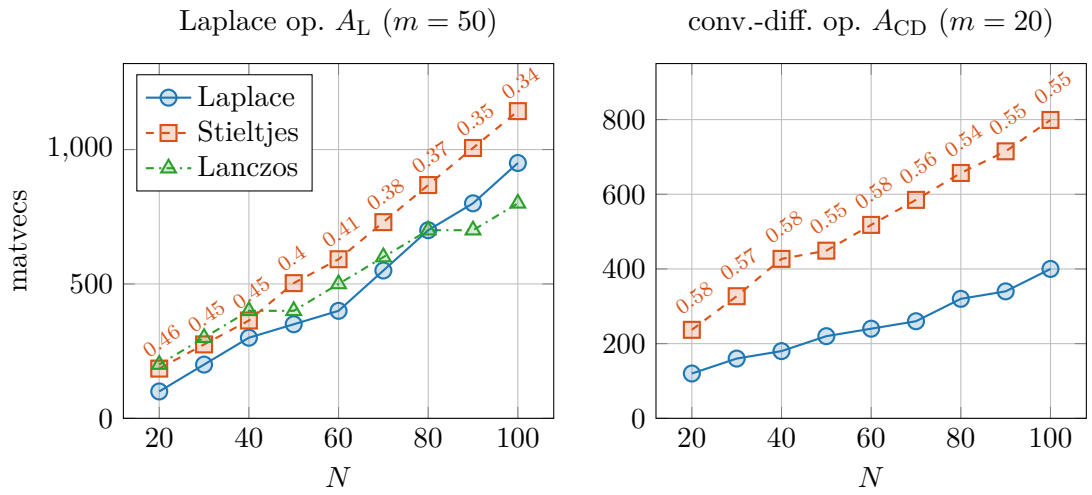


Figure 4.1.: Number of matrix-vector products ("matvecs") for approximating $A^{-3/2}\boldsymbol{b}$. "Laplace" denotes Algorithm 4. "Stieltjes" is the combination of `funm_quad` with CG (left) or GMRES (right). The small numbers above the dashed line indicate which fraction of the overall number of matrix-vector products is computed in the first phase, i.e., in the CG or GMRES method. "Lanczos" denotes the two-pass Lanczos method. The restart length is $m$.

matrix-vector products using directly the Laplace transform (our algorithm) than the detour via the Stieltjes function (CG/GMRES and `funm_quad`). It turns out that our algorithm (for $f(s)$) and `funm_quad` (for $h(s) = f(s)\cdot s$) perform quite similarly and that the difference in matrix-vector products is mostly due to the CG or GMRES method. To illustrate this, we include the ratio of the matrix-vector products that are caused by CG or GMRES to the total amount. They are listed above each data point in Fig. 4.1. One interpretation is thus that we can save some matrix-vector products by unifying the initial solving of a linear system with the evaluation of the matrix function into a single algorithm. The two-pass Lanczos method also needs more matrix-vector products than Algorithm 4 for $N \leq 70$. However, this is not the case for $N > 70$ so it might not be a systematic observation.

In Fig. 4.2, we show the convergence curves for $N = 100$ (after the initial CG/GMRES phase). They further illustrate that Algorithm 4 and `funm_quad` by themselves behave similarly to each other while the two-pass Lanczos method exhibits a better asymptotic rate of convergence.

The overall smaller number of matrix-vector products for our algorithm compared to the "Stieltjes" case is not caused by a general loss of the achieved accuracy. We back this up by plotting the achieved relative error norms in Fig. 4.3.

We now want to illustrate that we implemented our algorithm efficiently in the sense that the computational cost is $\mathcal{O}(n)$, where $n = N^3$ is the size of the matrix $A$. We plot

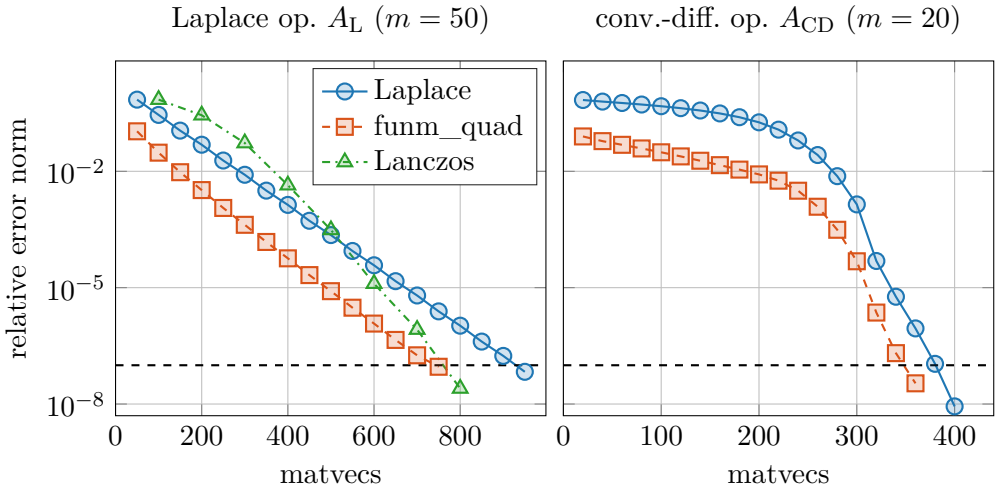


Figure 4.2.: Convergence curves for $A^{-3/2}\boldsymbol{b}$ with Algorithm 4 ("Laplace") or the two-pass Lanczos method ("Lanczos") and for $A^{-1/2}\boldsymbol{c}$ where $\boldsymbol{c} = A^{-1}\boldsymbol{b}$, with `funm_quad`. $N = 100$. The restart length is $m$.

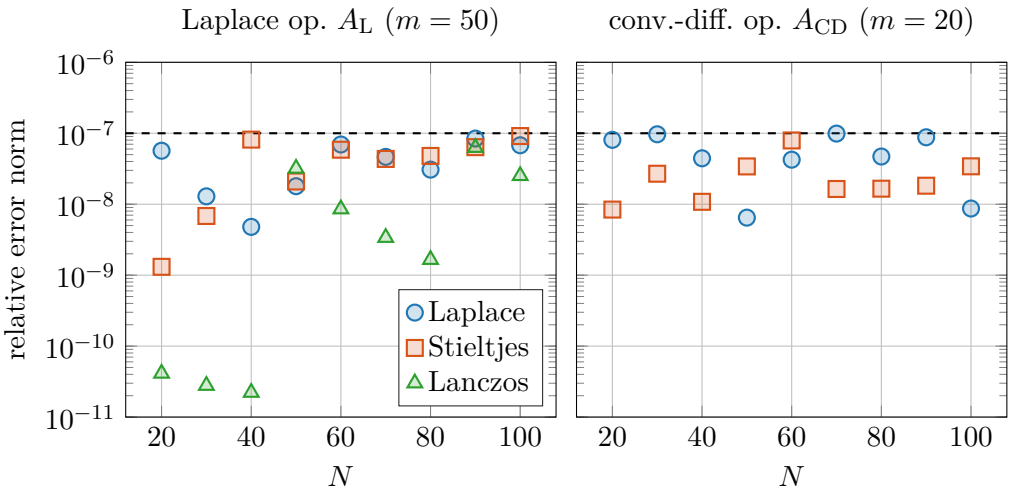


Figure 4.3.: Accuracy at termination when approximating $A^{-3/2}\boldsymbol{b}$. "Laplace" denotes Algorithm 4. "Stieltjes" is the combination of `funm_quad` with CG (left) or GMRES (right). "Lanczos" denotes the two-pass Lanczos method. The restart length is $m$.

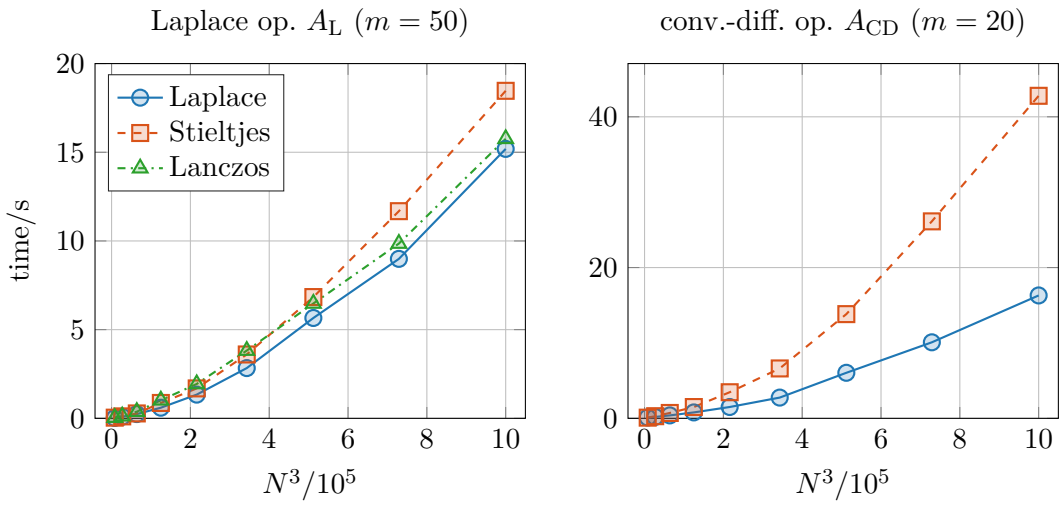


Figure 4.4.: Execution times when approximating $A^{-3/2}\boldsymbol{b}$ for varying matrix sizes $N^3$. "Laplace" denotes Algorithm 4. "Stieltjes" is the combination of `funm_quad` with CG (left) or GMRES (right). "Lanczos" denotes the two-pass Lanczos method. The restart length is $m$.

the execution times for the various sizes in Fig. 4.4. We see that the resulting curve becomes linear for increasing $n$ (neglecting some inaccuracy in measurement), which confirms our claim. We also observe that the execution time correlates with the number of matrix-vector products: Algorithm 4 needs less time than `funm_quad` in all examined cases. Of course, execution times are highly dependent on the implementation, so one should not derive general statements from Fig. 4.4.

Lastly, we want to confirm that our choices for the restart length $m$ are not only common but indeed appropriate. In Fig. 4.5, we show the execution times for varying values of $m$ for $N = 100$. A larger restart length can improve the rate of convergence, which should yield a smaller execution time. This can be indeed observed in the Hermitian case $A_{\mathrm{L}}$. However, in the non-Hermitian case $A_{\mathrm{CD}}$, a larger restart length increases the cost for the Arnoldi process: The next basis vector for the Krylov subspace needs to be orthogonalized against all previous ones, so one wants to keep the number of basis vectors (i.e., the restart length) small.

### 4.3.2. Fractional diffusion processes on graphs

Next, we choose a more practically relevant Laplace transform (based on [L, Section 6.2]). In fractional diffusion processes, one is interested in

$$\exp(-\tau L^{\alpha})\boldsymbol{b}, \quad \alpha \in (0,1),\ \tau > 0,$$

see, e.g., [11]. The matrix $L$ here is a graph Laplacian, i.e., given an undirected graph $G$, we have $L = D - A$, where $D$ is the diagonal matrix of degrees and $A$ is the adjacency

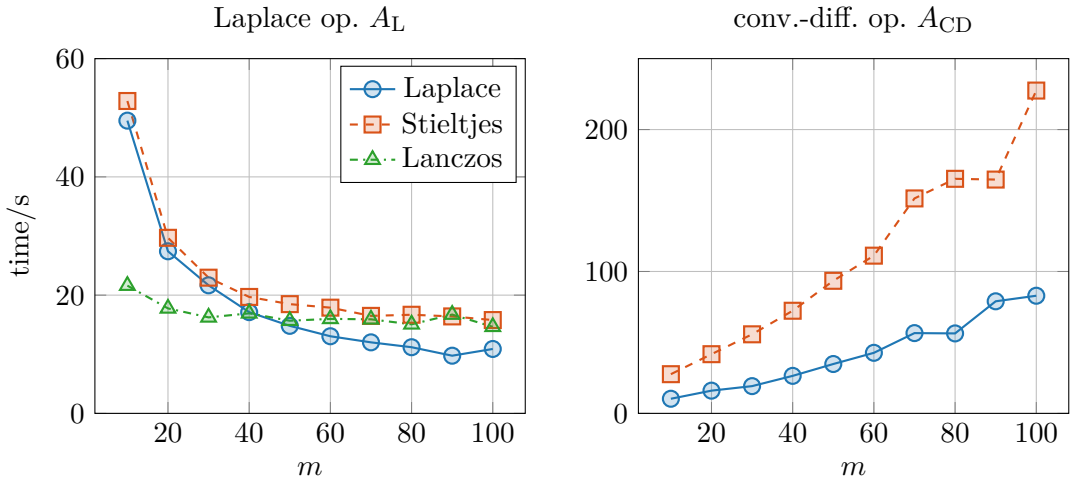


Figure 4.5.: Execution times when approximating $A^{-3/2}\boldsymbol{b}$ for varying restart lengths $m$ and $N = 100$. "Laplace" denotes Algorithm 4. "Stieltjes" is the combination of `funm_quad` with CG (left) or GMRES (right). "Lanczos" denotes the two-pass Lanczos method. Note that for two-pass Lanczos, $m$ specifies that the stopping criterion is checked every $m$ iterations.

matrix. We choose $\alpha = \frac{1}{2}$, which leads to

$$f(s) = \exp(-\tau\sqrt{s}) = \frac{\tau}{2\sqrt{\pi}}\mathcal{L}\{\frac{\exp(-\tau^2/(4t))}{t^{3/2}}\}(s), \quad \alpha\{\hat{f}\} = 0,$$

see, e.g., [25, Table of Laplace Transforms, No. 49].

**Remark 4.22.** As a generalization of Example 2.37, the function $s^\alpha$ with $\alpha \in (0,1)$ is a complete Bernstein function, see [80, Section 16.2, No. 1]. On one hand, this implies that $f(s) = \exp(-\tau s^\alpha)$ is a completely monotone function by Theorem 3.7 in [80]. On the other hand, every completely monotone function can be expressed by a Laplace transform if one allows more general measures than the Lebesgue measure ([80, Theorem 1.4]). It follows that $\exp(-\tau s^\alpha)$ is a Laplace transform for all values of $\alpha \in (0,1)$. We choose $\alpha = \frac{1}{2}$ because a closed-form expression for the inverse Laplace transform $\hat{f}$ is available for this value.

We want to use `funm_quad` again for comparison: The function $h(s) \coloneqq (f(s)-1)s^{-1}$ has the integral representation

$$h(s) = \frac{f(s)-1}{s} = -\int_{-\infty}^{0} \frac{1}{t-s}\frac{\sin(\tau\sqrt{-t})}{\pi t}\,\mathrm{d}t,$$

see [28, Example 1]. Theorem 2.67 gives an integral representation of the error functions

Table 4.1.: Number of nodes and edges of the largest connected components. The graphs were obtained from [24].

| Name | nodes | edges |
|---|---|---|
| usroads-48 | 126 146 | 323 900 |
| loc-Gowalla | 196 591 | 1 900 654 |
| dblp-2010 | 226 413 | 1 432 920 |
| com-Amazon | 334 863 | 1 851 744 |

for this kind of integral.[10] Furthermore, simply switching the sign of the integration variable yields

$$h(s) = \int_0^\infty \frac{\rho(t)}{s+t}\,\mathrm{d}t, \quad \rho(t) = -\frac{\sin(\tau\sqrt{t})}{\pi t}.$$

While $h(s)$ is not a Stieltjes function since $\rho(t) \not\geq 0$, we can still use `funm_quad` to evaluate $f(L)\boldsymbol{b} = h(L)(L\boldsymbol{b}) + \boldsymbol{b}$.

We choose the largest connected components of four real-world graphs from the SuiteSparse Matrix Collection [24] to construct the Laplacian matrix $L$. The graphs are listed in Table 4.1 together with the number of nodes (i.e., the size of $L$) and the number of edges (i.e., the number of non-zero, non-diagonal elements of $L$) of their largest connected component. Moreover, we chose $\boldsymbol{b}$ to be $\boldsymbol{e}_1$ orthogonalized against $[1,\dots,1]^\mathsf{T} \in \mathbb{R}^n$, see Remark 4.23 below. We present the convergence curves for $\tau = 1$ in Fig. 4.6. We observe that computing $f(L)\boldsymbol{b}$ with Algorithm 4 needs significantly fewer matrix-vector products than via `funm_quad`; in one case even less than half. The two-pass Lanczos method performs even better except for the graph "dblp-2010".

**Remark 4.23.** Note that our matrices are positive semidefinite, i.e., they have 0 as an eigenvalue. This means that, in principle, the matrix $h(L)$ is undefined. Moreover, $\alpha\{\hat{f}\} = 0$, so the requirement

$$\alpha\{\hat{f}\} < \min_{s\in\mathcal{W}(L)} \mathrm{Re}(s)$$

in Theorem 4.2 is not fulfilled. From this point of view, neither Algorithm 4 nor `funm_quad` should be applicable. However, we can effectively remove the zero eigenvalue by so-called *desingularization*, which we shortly explain in the following.

We used here *implicit desingularization* as described in [11, Section 5.2]. We orthogonalized the vector $\boldsymbol{b} = \boldsymbol{e}_1$ against the eigenvector $\boldsymbol{v} = [1,\dots,1]^\mathsf{T} \in \mathbb{R}^n$ corresponding to the eigenvalue 0. This resulted in the new vector

$$\tilde{\boldsymbol{b}} = \boldsymbol{b} - \frac{\boldsymbol{v}^\mathsf{T}\boldsymbol{b}}{n}\boldsymbol{v}.$$

[10] Note that Theorem 2.67 does not guarantee that the integral representation exists; in our case, this is guaranteed by [81, Proposition 3.8], however.

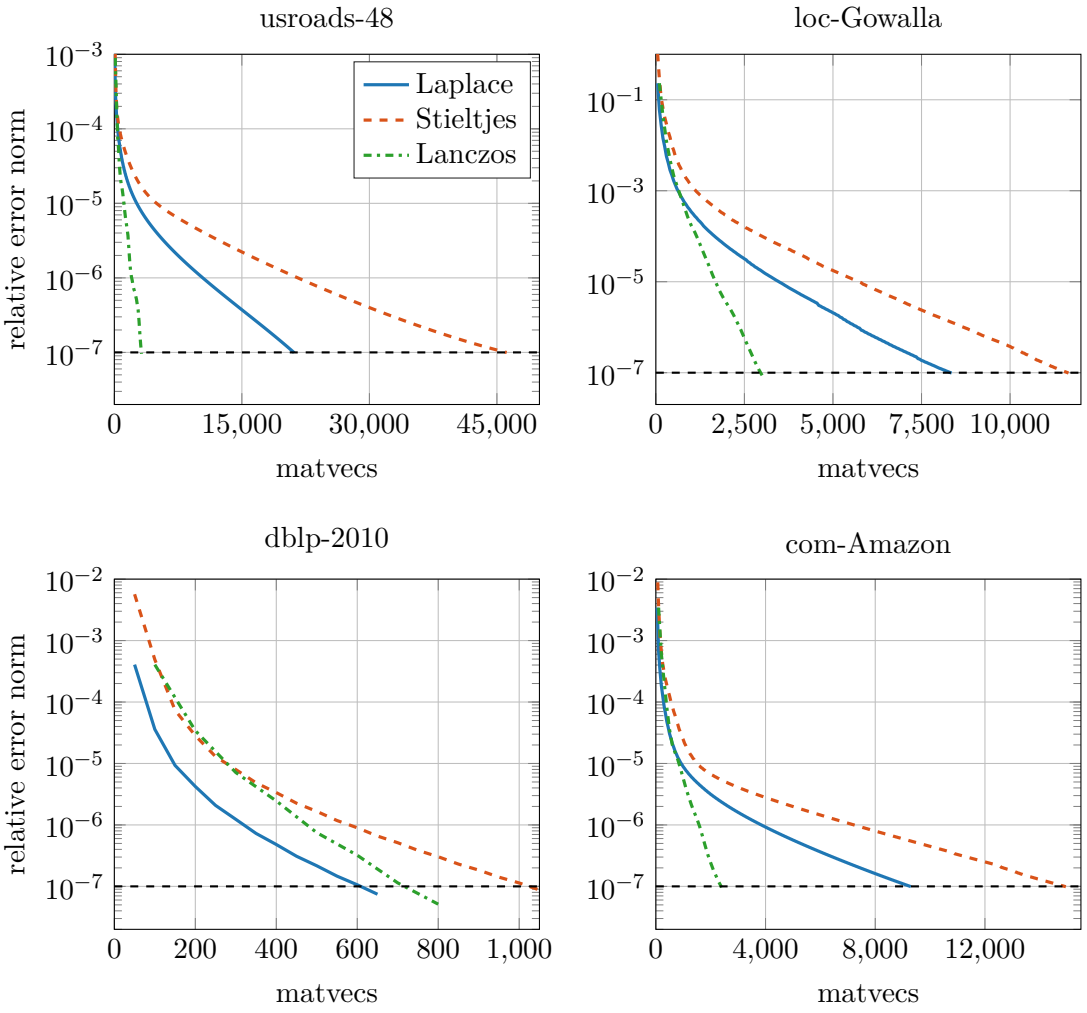


Figure 4.6.: Convergence curves for approximating $\exp(-\sqrt{L})\boldsymbol{b}$. "Laplace" denotes Algorithm 4. "Stieltjes" refers to the final error obtained by using `funm_quad` for $h(L)(L\boldsymbol{b})$ in $\exp(-\sqrt{L})\boldsymbol{b} = (h(L)L + I)\boldsymbol{b}$. "Lanczos" denotes the two-pass Lanczos method. The restart length is $m = 50$.

It can be shown (see [11, Theorem 5.1]) that a Krylov subspace method for $f(L)\tilde{\boldsymbol{b}}$ then yields the same result as for the projected expression $Qf(Q^{\mathsf{T}}LQ)(Q^{\mathsf{T}}\boldsymbol{b})$, where the columns of $Q \in \mathbb{C}^{n\times(n-1)}$ form an orthonormal basis for the complement of $\operatorname{span}(\boldsymbol{v})$. In other words, by ensuring that the right-hand side is orthogonal to the nullspace of the matrix, we effectively solve the problem within its range instead of the whole space. The eigenvalue 0 is not present in this subspace.

Note that one can easily retrieve $f(L)\boldsymbol{b}$ after desingularization: If one has evaluated

$$f(L)\tilde{\boldsymbol{b}} = f(L)\boldsymbol{b} - \frac{\boldsymbol{v}^{\mathsf{T}}\boldsymbol{b}}{n} f(L)\boldsymbol{v},$$

one just needs $f(L)\boldsymbol{v}$, which simplifies to $f(0)\boldsymbol{v}$ since $\boldsymbol{v}$ is an eigenvector of $L$.

### 4.3.3. Gamma function

As an example of a two-sided Laplace transform, we present experiments involving the gamma function

$$f(s) = \Gamma(s) = \int_0^\infty x^{s-1}\exp(-x)\,\mathrm{d}x, \quad \operatorname{Re}(s) > 0,$$

cf. [L, Section 6.3]. The computation of the matrix gamma function has recently received attention in [19, 72]. To show that $\Gamma(s)$ is indeed a two-sided Laplace transform, we use the fact that $x = \exp(\log(x))$ for $x > 0$ and then the transformation $t = -\log(x)$. This results in

$$\begin{aligned}\Gamma(s) &= \int_0^\infty \exp\big((s-1)\log(x)\big)\exp\big(-\exp(\log(x))\big)\,\mathrm{d}x \\ &= \int_{-\infty}^\infty \exp(-st)\exp\big(-\exp(-t)\big)\,\mathrm{d}t \\ &= \mathcal{L}\{\exp(-\exp(-t))\}(s) + \mathcal{L}\{\exp(-\exp(t))\}(-s),\end{aligned}$$

which is the two-sided Laplace transform of $\hat{f}(t) = \exp(-\exp(-t))$. For the region of existence, we have $\alpha\{\hat{f}\} = 0$ by Lemma 2.6 since $\omega\{\hat{f}\} = 0$ and

$$\mathcal{L}\{\hat{f}\}(0) = \int_0^\infty \exp(-\exp(-t))\,\mathrm{d}t = \int_0^1 \frac{\exp(-x)}{x}\,\mathrm{d}x = \infty.$$

Moreover, $\alpha\{\hat{f}(-t)\} \le \omega\{\hat{f}(-t)\} = -\infty$. In summary, the gamma function $\Gamma(s)$ can be represented by a two-sided Laplace transform in the half-plane $\operatorname{Re}(s) > 0$.

We implemented Algorithm 4.5 combined with Algorithm 4.1 of [19]. This way we can compute $\Gamma(A)$ directly and thus determine the error for $\Gamma(A)\boldsymbol{b}$ in our experiments. However, this approach scales like $\mathcal{O}(n^3)$, where $n$ is the size of $A$, so it becomes

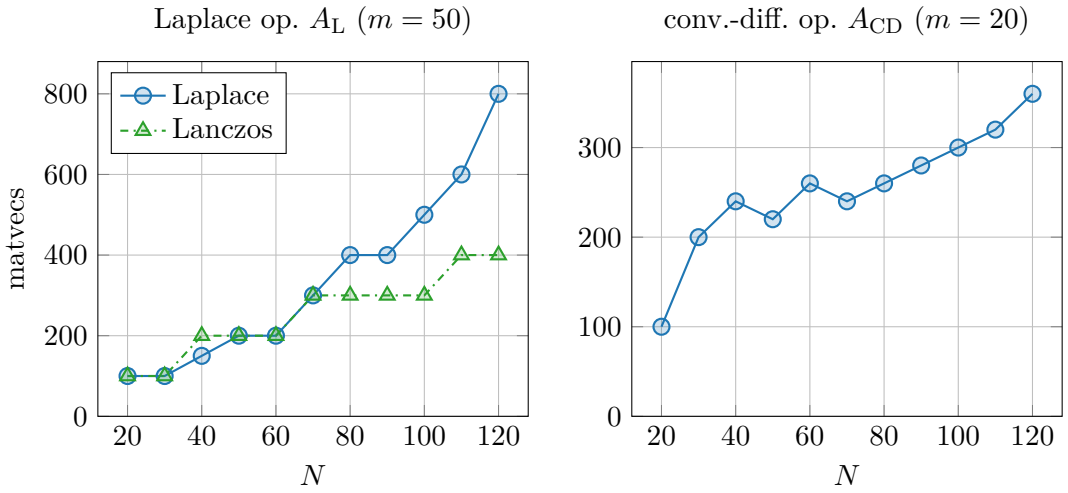


Figure 4.7.: Number of matrix-vector products for approximating $\Gamma(A)\boldsymbol{b}$. "Laplace" denotes Algorithm 4. "Lanczos" denotes the two-pass Lanczos method. The restart length is $m$.

prohibitively expensive for large values of $n$. We restrict ourselves consequently to the 2D versions of the matrices $A_{\mathrm{L}}$ and $A_{\mathrm{CD}}$ from Section 4.3.1, i.e., we use

$$A_{\mathrm{L}} = A_1 \oplus A_1 \in \mathbb{R}^{N^2 \times N^2},$$
$$A_{\mathrm{CD}} = \epsilon h^{-2} A_{\mathrm{L}} + h^{-1} A_2 \oplus A_2^{\mathsf{T}} \in \mathbb{R}^{N^2 \times N^2}.$$

Both matrices are again positive definite, so we can apply our Algorithm 4 for both $\mathcal{L}\{\hat{f}\}(A)\boldsymbol{b}$ and $\mathcal{L}\{\hat{f}(-t)\}(-A)\boldsymbol{b}$. We choose $\boldsymbol{b} = [1, \ldots, 1]^{\mathsf{T}} \in \mathbb{R}^{N^3}$ and let $N$ vary from 20 to 120, which results in $n$ varying from 400 to 14400.

The number of required matrix-vector products and the achieved accuracy are plotted in Figs. 4.7 and 4.8. While the two-pass Lanczos method requires fewer matrix-vector products for larger sizes of $A_{\mathrm{L}}$, we conclude that our algorithm can be employed for two-sided Laplace transforms.

### 4.3.4. Square root

The first complete Bernstein function we treat is the square root

$$f(s) = \sqrt{s} = \frac{1}{2\sqrt{\pi}} \int_0^\infty (1 - \exp(-ts)) t^{-3/2} \, \mathrm{d}t,$$

see Example 2.37. We base our discussion on [L, Section 6.4]. The action of the square root $f(A)\boldsymbol{b}$ has several applications including machine learning [76], sampling from Gaussian Markov random fields [59] and preconditioning [7].

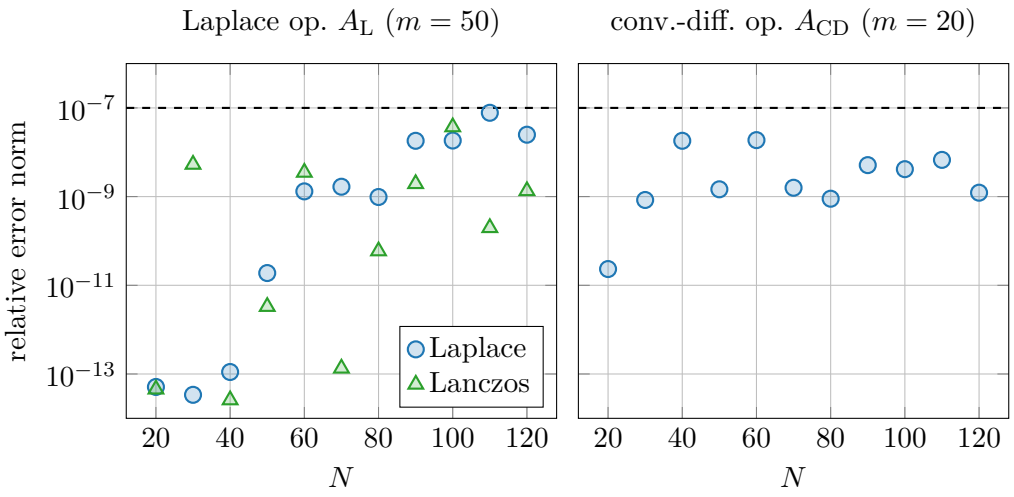


Figure 4.8.: Accuracy at termination when approximating $\Gamma(A)\boldsymbol{b}$. "Laplace" denotes Algorithm 4. "Lanczos" denotes the two-pass Lanczos method. The restart length is $m$.

We apply Theorem 2.7 to determine $\alpha\{t\hat{f}(t)\} = \alpha\{t^{-1/2}\}$: We know that

$$\int_0^t z^{-1/2}\,\mathrm{d}z = 2\sqrt{t}$$

and $\omega\{\sqrt{t}\} = 0$. It follows that $\alpha\{t\hat{f}(t)\} = 0$. This means we can compute the action of the square root for positive definite matrices using Algorithm 4, see Lemma 4.10. For comparison, the package `funm_quad` can also be used: We evaluate $h(A)\boldsymbol{c} = f(A)\boldsymbol{b}$, where $h(s) = s^{-1/2}$ is a Stieltjes function (see Example 2.30) and $\boldsymbol{c} = A\boldsymbol{b}$.

We choose again the 3D versions of the Laplace operator $A_{\mathrm{L}}$ and the convection-diffusion operator $A_{\mathrm{CD}}$ from Section 4.3.1 and $\boldsymbol{b} = [1, \ldots, 1]^{\mathsf{T}} \in \mathbb{R}^{N^3}$. The number of matrix-vector products and the achieved accuracy are plotted in Figs. 4.9 and 4.10. While the effect is not as strong in the non-Hermitian case, we see once again that Algorithm 4 needs fewer matrix-vector products than the detour via `funm_quad`. This time, it even needs fewer products than the two-pass Lanczos method (except for $N = 90$). Note, however, that the difference is usually only $m$ products. As only multiples of $m$ are possible for the reported number of products, this might not be significant.

### 4.3.5. Entropy

Let $A \in \mathbb{C}^{n\times n}$ be a density matrix, i.e., $A$ is Hermitian positive semi-definite with $\mathrm{trace}(A) = 1$. The *(von Neumann) entropy* of $A$ is then given by $\mathrm{trace}(-\log(A)A)$. As is mentioned in [10], the entropy appears in several fields including quantum statistical mechanics and network science, see the references in [10]. One often approximates it

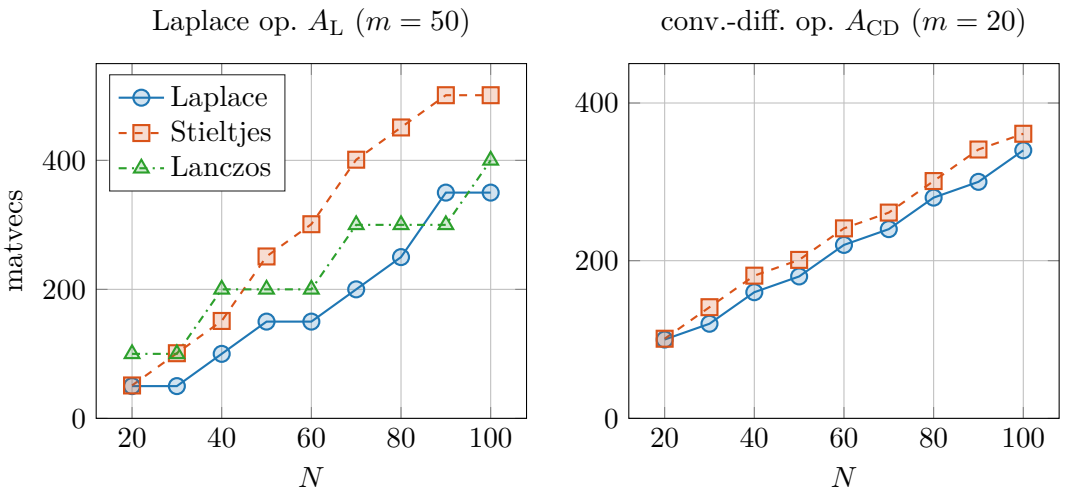


Figure 4.9.: Number of matrix-vector products for approximating $A^{1/2}\boldsymbol{b}$. "Laplace" denotes Algorithm 4. "Stieltjes" is `funm_quad` for $A^{-1/2}\boldsymbol{c}$ with $\boldsymbol{c} = A\boldsymbol{b}$. "Lanczos" denotes the two-pass Lanczos method. The restart length is $m$.

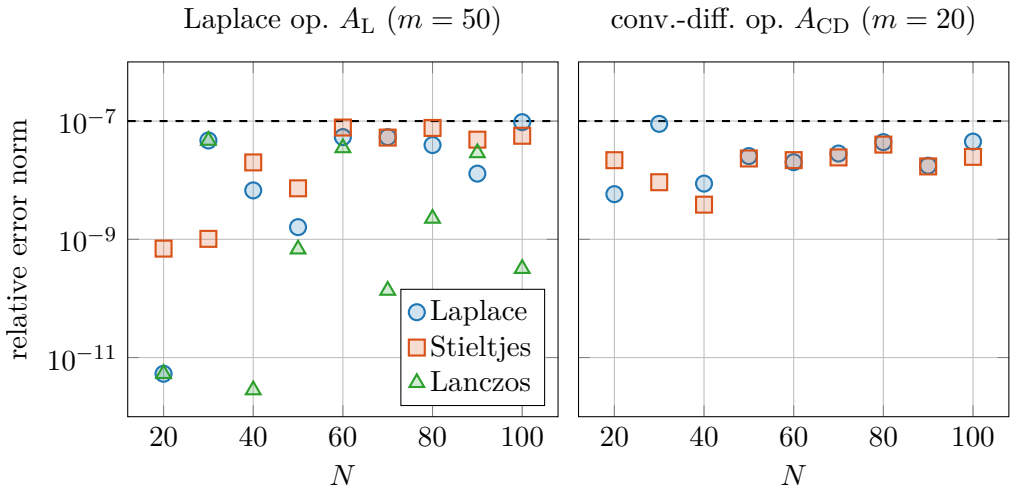


Figure 4.10.: Accuracy at termination when approximating $A^{1/2}\boldsymbol{b}$. "Laplace" denotes Algorithm 4. "Stieltjes" is `funm_quad` for $A^{-1/2}\boldsymbol{c}$ with $\boldsymbol{c} = A\boldsymbol{b}$. "Lanczos" denotes the two-pass Lanczos method. The restart length is $m$.

by

$$\operatorname{trace}(-\log(A)A) \approx -\sum_{i=1}^{\ell} \boldsymbol{v}_i^{\mathsf{T}} \log(A) A \boldsymbol{v}_i$$

with so-called *probing vectors* $\boldsymbol{v}_i$ and $\ell \ll n$. If we write $\boldsymbol{b} = A\boldsymbol{v}_i$, then we can easily evaluate the term $\boldsymbol{v}_i^{\mathsf{T}} \log(A)\boldsymbol{b}$ with our restart method by applying it to

$$f(A)\boldsymbol{b} = \log(A)\boldsymbol{b}.$$

This is possible because the logarithm is a complete Bernstein function with

$$\log(s+1) = \int_0^\infty (1 - \exp(-ts))\hat{f}(t)\,\mathrm{d}t, \quad \hat{f} = \exp(-t)t^{-1}$$

according to [80, Section 16.4, No. 26].

Note that the integral representation above gives $\log(A + I)\boldsymbol{b}$ whereas we want $\log(A)\boldsymbol{b}$. One might be tempted to define $M = A - I$ and then compute $\log(M + I)\boldsymbol{b} = \log(A)\boldsymbol{b}$. However, the matrix $M$ has negative eigenvalues. Theoretically, this is not a problem because $\alpha\{t\exp(-t)t^{-1}\} = \alpha\{\exp(-t)\} = -1$. Moreover, $M$ can have $-1$ as an eigenvalue only if $A$ has 0 as an eigenvalue. We use implicit desingularization as in Remark 4.23 since $\boldsymbol{b} = A\boldsymbol{v}_i$, thus 0 is effectively not present. In practice, however, negative eigenvalues can lead to numerically difficult integrals.

Luckily, the exponential function $\exp(-t)$ represents a shift in the context of Laplace transforms. Indeed, since

$$AV_m = V_m H_m + h_{m+1,m}\boldsymbol{v}_{m+1}\boldsymbol{e}_m^{\mathsf{T}} \implies (A - I)V_m = V_m(H_m - I) + h_{m+1,m}\boldsymbol{v}_{m+1}\boldsymbol{e}_m^{\mathsf{T}},$$

we have

$$\begin{aligned}
\mathcal{L}\{\hat{f}^{(2)}\}(M) &= \int_0^\infty \exp(-t(A-I)) \int_0^\infty \hat{f}^{(1)}(t+\tau)\boldsymbol{e}_m^{\mathsf{T}} \exp(-\tau(H_m^{(1)} - I))\boldsymbol{e}_1\,\mathrm{d}\tau\,\mathrm{d}t \\
&= \int_0^\infty \exp(-tA) \int_0^\infty (t+\tau)^{-1}\boldsymbol{e}_m^{\mathsf{T}} \exp(-\tau H_m^{(1)})\boldsymbol{e}_1\,\mathrm{d}\tau\,\mathrm{d}t \\
&= \mathcal{L}\{\hat{F}^{(2)}\}(A),
\end{aligned}$$

where $\hat{F}(t) = \hat{f}(t)\exp(t) = t^{-1}$ and $\hat{F}^{(2)}$ follows formally by restarting $\mathcal{L}\{\hat{F}\}(A)\boldsymbol{b}$. In other words, the term $\exp(-t)$ in $\hat{f}$ and the shift $-I$ for our matrix cancel each other out after the first cycle. Before that, i.e., in the first cycle, we do not need the integral representation since we restricted ourselves to Hermitian $A$. Accordingly, we can work directly with $\hat{F} = t^{-1}$ and $A$.

We use the implementation that is meant for Laplace transforms as explained in Section 4.2.4, i.e., we pass $\hat{f}(t) \approx -(t+\epsilon)^{-1}$ with $\epsilon = \varepsilon_{\mathrm{q}} = 10^{-10}$. As test matrices, we choose the graph Laplacians of Section 4.3.2 but normalize by dividing by their traces so that the resulting matrices are density matrices, i.e., $A = \operatorname{trace}(L)^{-1}L$. We also choose $\boldsymbol{e}_1$ as probing vector, which means $\boldsymbol{b} = A\boldsymbol{e}_1$. The results are plotted in Fig. 4.11.

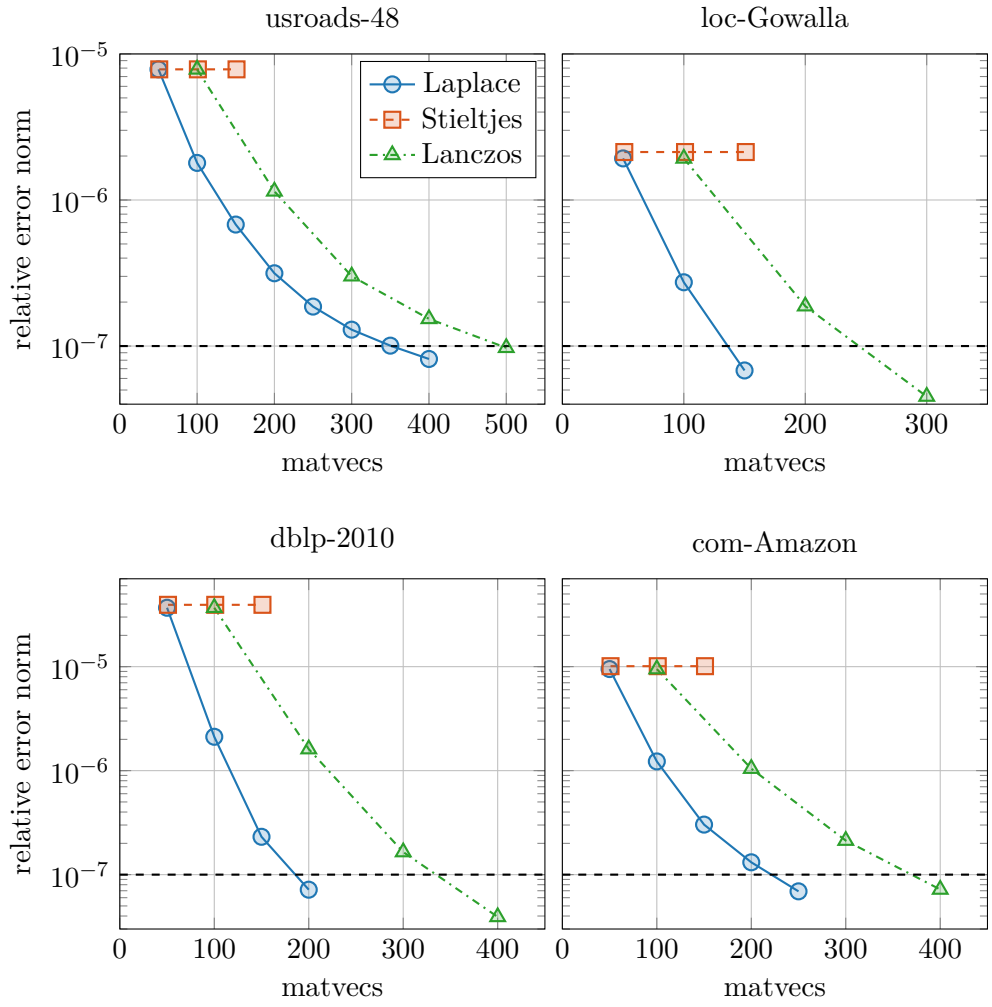


Figure 4.11.: Convergence curves for approximating $\log(A)\boldsymbol{b}$. "Laplace" denotes Algorithm 4. "Stieltjes" refers to the final error obtained by using `funm_quad` for $h(M)\boldsymbol{c}$ with $h(s) = \log(s)s^{-1}$, $M = A - I$ and $\boldsymbol{c} = M\boldsymbol{b}$. "Lanczos" denotes the two-pass Lanczos method. The restart length is $m = 50$.

In the previous examples, our algorithm often needed more matrix-vector products than the two-pass Lanczos method. This time, however, we observe that the convergence rate does not worsen much when restarting: Except for usroads-48, the $k$th approximation of Algorithm 4 is only marginally worse than the $k$th approximation of the two-pass Lanczos method. As the latter needs to perform every matrix-vector product twice, Algorithm 4 thus requires only around half as many matrix-vector products. We also observe that the approximation $t^{-1} \approx (t+\epsilon)^{-1}$ did not prevent Algorithm 4 from reaching the target accuracy.

We tried using `funm_quad` as well: The function $h(s) = \log(s+1)s^{-1}$ is a Stieltjes function and we can write $\log(A)\boldsymbol{b} = h(M)(M\boldsymbol{b})$. We observe in Fig. 4.11 that the first cycle yields an approximation with a similar error to the other two methods. After that, it fails to improve it, however. A numerical investigation by Marcel Schweitzer (personal communication, University of Wuppertal, Feb 2023) revealed that this is mostly due to the combination of the spectrum of $M$ and the quadrature in `funm_quad`: Since

$$\frac{\log(s+1)}{s} = \int_1^\infty \frac{1}{t(t+s)}\,\mathrm{d}t,$$

the error function is

$$f^{(k+1)}(s) = (-1)^k \Big(\prod_{j=1}^{k} h^{(j)}_{m+1,m}\Big) \int_1^\infty t^{-1} \Big(\prod_{j=1}^{k} \psi^{(j)}_m(t)\Big)(s+t)^{-1}\,\mathrm{d}t$$

with $\psi^{(j)}_m(t) = \boldsymbol{e}_m^\mathsf{T}(H^{(j)}_m + tI)^{-1}\boldsymbol{e}_1$, cf. Theorem 2.68. The matrices $A$ have eigenvalues only close to 0 and so the eigenvalues of $M$ and subsequently of $H^{(j)}_m$ are all close to $-1$. This means that $\psi^{(j)}_m(t)$, $(t+s)^{-1}$ (with $s = H^{(k+1)}_m$) and consequently the integrand have significant values only for $t \approx 1$. The quadrature rule in `funm_quad` does not detect this and thus yields a value close to 0 for the integral.[11] This results in updates too small to change the error norm significantly.

## 4.4. Error bounds

In Section 4.1, we developed a new representation for the error of restarted Arnoldi for Laplace transforms. We also explained that it can be used to construct an efficient restarting algorithm in Section 4.2. However, we do not know yet whether the approximation $\boldsymbol{d}^{(k)}_m$ of our algorithm will become arbitrarily close to $f(A)\boldsymbol{b}$ if we just invest enough restart cycles, i.e., whether our algorithm converges.

Therefore, we present a new a priori error bound for Laplace transforms. A classical result (see Lemma 2.6) about Laplace transforms is that $\mathcal{L}\{\hat{f}\}(s)$ with $\hat{f} \in L^1_{\mathrm{loc}}(\mathbb{R}^+_0)$ exists if $\omega\{\hat{f}\} < \mathrm{Re}(s)$. That is, if there are $T$, $c$ and $\omega$ large enough such that

$$|\hat{f}(t)| \leq c\exp(t\omega)$$

[11] Before computing the Gauß-Laguerre quadrature rule, `funm_quad` applies a transformation to the integral, see [40]. This transformation does not change the argument, however.

for almost all $t \geq T$, then $\mathcal{L}\{\hat{f}\}(s) < \infty$ for every $s$ with $\omega < \operatorname{Re}(s)$. Our bound is based on the following idea: We split the integral $\mathcal{L}\{\hat{f}\}(s)$ at this $T$, i.e.,

$$\begin{aligned}\mathcal{L}\{\hat{f}\}(s) &= \int_0^\infty \exp(-ts)\hat{f}(t)\,\mathrm{d}t \\ &= \int_0^T \exp(-ts)\hat{f}(t)\,\mathrm{d}t + \int_T^\infty \exp(-ts)\hat{f}(t)\,\mathrm{d}t \\ &= \mathcal{L}\{\hat{f}\chi_{(0,T)}\}(s) + \mathcal{L}\{\hat{f}\chi_{(T,\infty)}\}(s).\end{aligned}$$

We consider three cases:

- The first case regards functions of the form $\mathcal{L}\{\hat{f}\chi_{(0,T)}\}(s)$. Such a function is an entire function, see Corollary 2.21. Thus, we expect superlinear convergence by Theorem 2.70. We derive explicit representations for the involved constants in Section 4.4.1.
- In Section 4.4.2, we consider functions $\mathcal{L}\{\hat{f}\}(s)$, where $\hat{f}(t)$ can be exponentially bounded for $t \geq 0$. Since $\mathcal{L}\{\exp(t\omega)\}(s) = (s-\omega)^{-1}$ (see Example 2.9), we should be able to apply knowledge about shifted linear systems. This case also includes functions $\mathcal{L}\{\hat{f}\chi_{(T,\infty)}\}(s)$, where $\hat{f}(t)$ can be bounded only for $t \geq T$.
- The more general case described above is obtained by combining the two cases. We present the main result in Section 4.4.3. Similarly to Theorem 4.2, it can be easily extended to complete Bernstein functions.

Afterward, we also shortly explain how an a posteriori error bound can be computed in Section 4.4.4. Before we start, we want to mention that under mild assumptions we can also show that the error cannot increase:

**Lemma 4.24** ([26, Remark 1])**.** *Let $A$ be Hermitian. Let $\hat{f}(t)$ be real and have constant sign. The error for the restarted Arnoldi method for $\mathcal{L}\{\hat{f}\}(s)$ satisfies*

$$\|\boldsymbol{\varepsilon}_m^{(k)}\|_2 \leq \|\boldsymbol{\varepsilon}_{m-1}^{(k)}\|_2$$

*for $k \geq 1$ and $m \geq 2$.*

*Proof.* The case $k = 1$ is proved in [26, Remark 1]. Thus, it suffices to show that $\hat{f}^{(k+1)}$ is real and has constant sign if the same holds for $\hat{f}^{(k)}$. The rest follows then by induction. By definition of $\hat{f}^{(k+1)}$,

$$\hat{f}^{(k+1)}(t) = \int_0^\infty \hat{f}^{(k)}(t+\tau)g^{(k)}(\tau)\,\mathrm{d}\tau,$$

we see that we need to investigate $g^{(k)}$. Because $A$ is Hermitian, we know that $g^{(k)}$ is real and has constant sign; we show this later in Lemma 4.28. Consequently, the same holds for $\hat{f}^{(k+1)}(t)$. □

### 4.4.1. A priori bound I: Finite integration interval

While we know from Theorem 2.70 that the error for $\mathcal{L}\{\hat{f}\chi_{(0,T)}\}(s)$ converges superlinearly, we want to include the dependence of $T$. We first describe an error bound for the exponential function in Lemma 4.25 and then integrate it in Lemma 4.26.

**Lemma 4.25** (cf. [30, Theorem 4.2])**.** *Let $\boldsymbol{y}_m^{(k)}(t)$ be the approximation to $\exp(-tA)\boldsymbol{b}$ from the restarted Arnoldi method. Let $\Gamma$ be a closed contour enclosing $\mathcal{W}(A)$. Denote by $\ell(\Gamma)$ its length, by $\operatorname{dist}(\Gamma, \mathcal{W}(A)) > 0$ its distance to $\mathcal{W}(A)$ and by $\xi = \min_{s\in\Gamma} \operatorname{Re}(s)$ its smallest real part. Then*

$$\|\exp(-tA)\boldsymbol{b} - \boldsymbol{y}_m^{(k)}(t)\|_2 \leq \|\boldsymbol{b}\|_2 C_0 \frac{(C_1 t)^{km}}{(km)!} \exp(-t\xi),$$

*where*

$$C_0 = \frac{\ell(\Gamma)}{2\pi \operatorname{dist}(\Gamma, \mathcal{W}(A))}, \quad C_1 = \max_{s\in\Gamma} |s| + \rho(A).$$

*Proof.* The statement is essentially [30, Theorem 4.2]. To see the dependence on $t$, we have to modify their proof. First note that $\boldsymbol{y}_m^{(k)}(t) = q_t(A)\boldsymbol{b}$ by Lemma 2.65. (As a reminder, the polynomial $q_t$ is the Hermite interpolating polynomial that interpolates $\exp(-ts)$ at the spectrum of $\Upsilon_{km}$ as given in Lemma 2.65.) This means we can write the error as a new matrix function $\exp(-tA)\boldsymbol{b} - \boldsymbol{y}_m^{(k)}(t) = f_t(A)\boldsymbol{b}$ with $f_t(s) = \exp(-ts) - q_t(s)$. This function is an entire function (since the exponential function and any polynomial are entire functions), so we can represent it using the Cauchy integral formula (see Definition 2.57) and obtain

$$\begin{aligned}
\|\exp(-tA)\boldsymbol{b} - \boldsymbol{y}_m^{(k)}(t)\|_2 &= \|\frac{1}{2\pi i} \int_\Gamma (sI - A)^{-1}\boldsymbol{b} f_t(s)\,\mathrm{d}s\|_2 \\
&\leq \frac{1}{2\pi} \int_\Gamma \|(sI - A)^{-1}\boldsymbol{b}\|_2 |f_t(s)|\,\mathrm{d}s \\
&\leq \frac{\|\boldsymbol{b}\|_2}{2\pi \operatorname{dist}(\Gamma, \mathcal{W}(A))} \int_\Gamma |f_t(s)|\,\mathrm{d}s.
\end{aligned}$$

We used $\|(sI - A)^{-1}\|_2 \leq \operatorname{dist}(\Gamma, \mathcal{W}(A))^{-1}$ from [30, Eq. (4.14)] for the last inequality. Now, assume that the Hermite interpolating polynomial $q_t(s)$ can be written as

$$q_t(s) = \sum_{j=1}^{km} \Delta_{[\theta_1,\ldots,\theta_j]}\{\exp(-ts)\} \cdot (s - \theta_1)\ldots(s - \theta_{j-1}),$$

where $\theta_j$ are the eigenvalues of $H_m^{(i)}$ for $i = 1, \ldots, k$ and $\Delta_{[\theta_1,\ldots,\theta_j]}\{\exp(-ts)\}$ are the corresponding divided differences. The fact that $q_t(s)$ can be represented in this form is proved later in Lemma 4.26. A well-known result regarding polynomial interpolation is that this implies

$$|f_t(s)| = |\Delta_{[\theta_1,\ldots,\theta_{km},s]}\{\exp(-ts)\}| \prod_{j=1}^{km} |s - \theta_j|,$$

see, e.g., [16, Chapter I, Theorem (14)]. The divided difference can be bounded by

$$\begin{aligned}|\Delta_{[\theta_1,\dots,\theta_{km},s]}\{\exp(-ts)\}| &\leq \frac{1}{(km)!}\max_{s\in\Omega}|\frac{\mathrm{d}^{km}}{\mathrm{d}s^{km}}\exp(-ts)| \\ &= \frac{t^{km}}{(km)!}\exp(-t\xi),\end{aligned}$$

where $\Omega$ is the convex hull of $\Gamma$, see, e.g., [56, Eq. (B.28)]. Since the eigenvalues $\theta_j$ are contained in $\mathcal{W}(A)$, we can bound the absolute value of the so-called nodal polynomial by

$$\prod_{j=1}^{km}|s-\theta_j| \leq \prod_{j=1}^{km}(|s|+|\theta_j|) \leq C_1^{km}.$$

The remaining integral can thus be bounded by

$$\begin{aligned}\int_\Gamma |f_t(s)|\,\mathrm{d}s &= \int_\Gamma |\Delta_{[\theta_1,\dots,\theta_{km},s]}\{\exp(-ts)\}| \prod_{j=1}^{km}|s-\theta_j|\,\mathrm{d}s \\ &\leq \frac{(C_1 t)^{km}}{(km)!}\exp(-t\xi)\int_\Gamma \mathrm{d}s = \frac{(C_1 t)^{km}}{(km)!}\exp(-t\xi)\ell(\Gamma). \qquad \square\end{aligned}$$

In the above proof, we assumed that the Hermite interpolating polynomial $q$ from Lemma 2.65 can be written in the form of Lemma 2.56. This is indeed true. While not a new result, it is usually only mentioned en passant, so we give a short proof:

**Lemma 4.26.** *The polynomial $q$ from Lemma 2.65 can be written as*

$$q(s) = \sum_{j=1}^{km}\Delta_{[\theta_1,\dots,\theta_j]}\{f\}\cdot(s-\theta_1)\dots(s-\theta_{j-1}),$$

*where $\theta_j$ are the eigenvalues of $H_m^{(i)}$ for $i=1,\dots,k$ and $\Delta_{[\theta_1,\dots,\theta_j]}\{f\}$ are the corresponding divided differences.*

*Proof.* Note that $\Upsilon_{km}$ is block lower triangular so its eigenvalues are the eigenvalues of its diagonal blocks $H_m^{(i)}$. Thus, we only need to show that the indices of $\theta_j$ coincide with their algebraic multiplicities because of Lemma 2.56. As $\Upsilon_{km}$ is also an upper Hessenberg matrix, we easily see that $\operatorname{rank}(\Upsilon_{km}-sI)\geq km-1$ by considering the sparsity of the columns; a zero entry on the subdiagonal is not possible as otherwise the Arnoldi process would have broken down, see Lemma 2.63. It follows that the geometric multiplicity of every eigenvalue of $\Upsilon_{km}$ is (at most and thus equal to) 1. Respectively, there is only one Jordan block for each distinct eigenvalue and the indices coincide with the algebraic multiplicity. $\square$

As mentioned before, we obtain an error bound for the first case, i.e., $\mathcal{L}\{\hat{f}\chi_{(0,T)}\}(s)$, by integrating over the result of Lemma 4.25:

**Lemma 4.27.** *Let $\hat{f} \in L^1([0,T])$. With the definitions of Lemma 4.25, the error of the restarted Arnoldi method for the function $\mathcal{L}\{\hat{f}\chi_{(0,T)}\}(s)$ satisfies*

$$\|\boldsymbol{\varepsilon}_m^{(k)}\|_2 \leq \|\boldsymbol{b}\|_2 C_0 \frac{(C_1 T)^{km}}{(km)!} \mathcal{L}\{|\hat{f}|\chi_{(0,T)}\}(\xi).$$

*Proof.* Starting from the definition of the error

$$\boldsymbol{\varepsilon}_m^{(k)} = \int_0^\infty (\exp(-tA)\boldsymbol{b} - \boldsymbol{y}_m^{(k)}(t))\hat{f}(t)\chi_{(0,T)}\,\mathrm{d}t,$$

we have

$$\|\boldsymbol{\varepsilon}_m^{(k)}\|_2 \leq \int_0^T \|\exp(-tA)\boldsymbol{b} - \boldsymbol{y}_m^{(k)}(t)\|_2 |\hat{f}(t)|\,\mathrm{d}t.$$

The lemma now follows easily by inserting Lemma 4.25 and noting that $t^{km} \leq T^{km}$. □

### 4.4.2. A priori bound II: Exponentially bounded integrand

Next, we consider the error for functions of the form $\mathcal{L}\{\hat{f}\}(s)$, where $|\hat{f}(t)| \leq c\exp(t\omega)$ for almost all $t \geq 0$. We use that the exponential bound is preserved after restarts, which we prove in Lemma 4.29. For this, we need an auxiliary result that we have already used in Lemma 4.24:

**Lemma 4.28.** *Let $A$ be Hermitian. Then $g^{(k)}(t) = (-1)^{m-1}|g^{(k)}(t)|$ for $k \geq 1$ and $t \geq 0$. In particular, $g^{(k)}(t)$ is real and has constant sign.*

*Proof.* Interpreting cycle $k$ as a separated start of the restarted Arnoldi method, we can set $\Upsilon_m = H_m^{(k)}$. Thus, Lemma 4.26 can be applied to $H_m^{(k)}$, which results in

$$g^{(k)}(\tau) = \boldsymbol{e}_m^\mathsf{T} \exp(-\tau H_m^{(k)})\boldsymbol{e}_1 = \sum_{j=1}^{m} \Delta_{[\theta_1,\ldots,\theta_j]}\{\exp(-\tau s)\} \cdot \boldsymbol{e}_m^\mathsf{T}\Big(\prod_{i=1}^{j-1}(H_m^{(k)} - \theta_i I)\Big)\boldsymbol{e}_1,$$

where $\theta_j$ are the eigenvalues of $H_m^{(k)}$. Note that $\boldsymbol{e}_m^\mathsf{T}(H_m^{(k)})^{j-1}\boldsymbol{e}_1$ and by extension the expressions

$$\boldsymbol{e}_m^\mathsf{T}\Big(\prod_{i=1}^{j-1}(H_m^{(k)} - \theta_i I)\Big)\boldsymbol{e}_1$$

vanish for $0 \leq j-1 < m-1$ because of the sparsity of the matrix $H_m^{(k)}$. Thus, $g^{(k)}$ simplifies to

$$\begin{aligned} g^{(k)}(\tau) &= \Delta_{[\theta_1,\ldots,\theta_m]}\{\exp(-\tau s)\} \cdot \boldsymbol{e}_m^\mathsf{T}\Big(\prod_{i=1}^{m-1}(H_m^{(k)} - \theta_i I)\Big)\boldsymbol{e}_1 \\ &= \Delta_{[\theta_1,\ldots,\theta_m]}\{\exp(-\tau s)\}\boldsymbol{e}_m^\mathsf{T}(H_m^{(k)})^{m-1}\boldsymbol{e}_1. \end{aligned}$$

Similarly, it is straightforward to see

$$\boldsymbol{e}_m^\mathsf{T}(H_m^{(k)})^{m-1}\boldsymbol{e}_1 = \prod_{j=1}^{m-1} h_{j+1,j}^{(k)},$$

which is positive. The divided difference, on the other hand, can be expressed as

$$\Delta_{[\theta_1,\dots,\theta_m]}\{\exp(-\tau s)\} = (-1)^{m-1}\frac{\tau^{m-1}}{(m-1)!}\exp(-\tau\xi)$$

for some $\xi \in \mathcal{W}(H_m^{(k)})$ due to $H_m^{(k)}$ being Hermitian, see [56, Eq. (B.26)]. It follows that $g^{(k)}(\tau) = (-1)^{m-1}|g^{(k)}(\tau)|$ since all terms besides $(-1)^{m-1}$ are non-negative. □

**Lemma 4.29.** *Let $|\hat{f}(t)| \leq c\exp(t\omega)$ for almost all $t \geq 0$. Let further $A$ be Hermitian and its smallest eigenvalue $\lambda_{\min} > \omega$. Then*

$$|\hat{f}^{(k+1)}(t)| \leq c\exp(t\omega)\prod_{j=1}^{k}|\psi_m^{(j)}(-\omega)|$$

*for all $t \geq 0$ with $\psi_m^{(j)}(-\omega) = \boldsymbol{e}_m^\mathsf{T}(H_m^{(k)} - \omega I)^{-1}\boldsymbol{e}_1$.*

*Proof.* The proof follows by induction. The case $k = 0$ is trivial by hypothesis. Now, assume that the equation holds for $k-1$, i.e.,

$$|\hat{f}^{(k)}(t)| \leq c\exp(t\omega)\prod_{j=1}^{k-1}|\psi_m^{(j)}(-\omega)|.$$

By definition of $\hat{f}^{(k+1)}$, it follows

$$\begin{aligned}|\hat{f}^{(k+1)}(t)| &\leq \int_0^\infty |\hat{f}^{(k)}(t+\tau)||g^{(k)}(\tau)|\,\mathrm{d}\tau \\ &\leq c\exp(t\omega)\prod_{j=1}^{k-1}|\psi_m^{(j)}(-\omega)|\int_0^\infty \exp(\tau\omega)|g^{(k)}(\tau)|\,\mathrm{d}\tau. \end{aligned} \tag{4.7}$$

The induction process and thus the proof concludes if we can bound the integral in Eq. (4.7) by $|\psi_m^{(k)}(-\omega)|$. We know that $g^{(k)}(\tau)$ has constant sign by Lemma 4.28. It follows

$$\begin{aligned}\int_0^\infty \exp(\tau\omega)|g^{(k)}(\tau)|\,\mathrm{d}\tau &= \left|\int_0^\infty \exp(\tau\omega)g^{(k)}(\tau)\,\mathrm{d}\tau\right| \\ &= \left|\int_0^\infty \exp(\tau\omega)\boldsymbol{e}_m^\mathsf{T}\exp(-\tau H_m^{(k)})\boldsymbol{e}_1\,\mathrm{d}\tau\right| \\ &= |\boldsymbol{e}_m^\mathsf{T}\mathcal{L}_\tau\{\exp(\tau\omega)\}(H_m^{(k)})\boldsymbol{e}_1| = |\psi_m^{(k)}(-\omega)|,\end{aligned}$$

where the Laplace transform exists for $H_m^{(k)}$ because $\lambda_{\min} > \omega$. □

We have already seen the term $\psi_m^{(1)}(0)$ when characterizing the residual of FOM in Lemma 2.74. We now relate the error for Laplace transforms to the error of FOM via the terms $\psi_m^{(j)}(-\omega)$. The reader is reminded that FOM coincides with CG for Hermitian positive definite matrices.

**Lemma 4.30.** *Let $A-\omega I$ be Hermitian positive definite and $\lambda_{\max}$ be the largest eigenvalue of $A$. Let $\boldsymbol{r}_m^{(k)}(-\omega)$ and $\boldsymbol{\epsilon}_m^{(k)}(-\omega)$ denote the residual and error of restarted FOM/CG applied to the matrix $A-\omega I$ and right-hand side $\boldsymbol{b}$. Then*

$$\|\boldsymbol{b}\|_2 \prod_{j=1}^{k} h_{m+1,m}^{(j)} |\psi_m^{(j)}(-\omega)| = \|\boldsymbol{r}_m^{(k)}(-\omega)\|_2 \leq \sqrt{\lambda_{\max}-\omega}\, \|\boldsymbol{\epsilon}_m^{(k)}(-\omega)\|_{A-\omega I}$$

*with $\psi_m^{(j)}(-\omega) = \boldsymbol{e}_m^{\mathsf{T}}(H_m^{(k)}-\omega I)^{-1}\boldsymbol{e}_1$ as before.*

*Proof.* The equality follows from combining the last equation on p. 1608 in [39] with Eq. (3.20) in [40]. It also follows from applying Lemma 2.74 recursively by noting that the elements $h_{m+1,m}^{(j)}$ are not influenced by the shift $-\omega I$. This can be seen by using the Arnoldi relation Eq. (2.3):

$$\begin{aligned}(A-\omega I)V_m^{(k)} &= AV_m^{(k)} - \omega V_m^{(k)} = V_m^{(k)}H_m^{(k)} + h_{m+1,m}^{(k)}\boldsymbol{v}_{m+1}^{(k)}\boldsymbol{e}_m^{\mathsf{T}} - \omega V_m^{(k)} \\ &= V_m^{(k)}(H_m^{(k)}-\omega I) + h_{m+1,m}^{(k)}\boldsymbol{v}_{m+1}^{(k)}\boldsymbol{e}_m^{\mathsf{T}}.\end{aligned}$$

The inequality follows from the simple relation $\boldsymbol{r}_m^{(k)}(-\omega) = (A-\omega I)\boldsymbol{\epsilon}_m^{(k)}(-\omega)$ and the bound $\|(A-\omega I)^{1/2}\|_2 \leq \lambda_{\max}-\omega$, i.e.,

$$\begin{aligned}\|\boldsymbol{r}_m^{(k)}(-\omega)\|_2 &= \|(A-\omega I)^{1/2}(A-\omega I)^{1/2}\boldsymbol{\epsilon}_m^{(k)}(-\omega)\|_2 \\ &\leq \sqrt{\lambda_{\max}-\omega}\, \|\boldsymbol{\epsilon}_m^{(k)}(-\omega)\|_{A-\omega I}.\end{aligned}$$

□

With this result, we now obtain the following error bound.

**Lemma 4.31.** *Let $\hat{f} \in L^1_{\mathrm{loc}}(\mathbb{R}_0^+)$ with $|\hat{f}(t)| \leq c\exp(t\omega)$ for almost all $t \geq 0$. Let $A$ be Hermitian and its smallest eigenvalue $\lambda_{\min} > \omega$. The error of the restarted Arnoldi method for $\mathcal{L}\{\hat{f}\}(s)$ satisfies*

$$\|\boldsymbol{\varepsilon}_m^{(k)}\|_2 \leq \|\boldsymbol{b}\|_2\, c \sqrt{\frac{\kappa(A-\omega I)}{\lambda_{\min}-\omega}}\, \|\boldsymbol{\epsilon}_m^{(k)}(-\omega)\|_{A-\omega I},$$

*where $\kappa(A-\omega I)$ is the condition number of $A-\omega I$ and $\boldsymbol{\epsilon}_m^{(k)}(-\omega)$ is the error of restarted CG for the matrix $A-\omega I$.*

*Proof.* We know that

$$\boldsymbol{\varepsilon}_m^{(k)} = \|\boldsymbol{b}\|_2(-1)^k\Big(\prod_{j=1}^{k} h_{m+1,m}^{(j)}\Big)\mathcal{L}\{\hat{f}^{(k+1)}\}(A)\boldsymbol{v}_{m+1}^{(k)}$$

from Corollary 4.8. We have $\|\exp(-tA)\boldsymbol{v}_{m+1}^{(k)}\| \leq \exp(-t\lambda_{\min})$ as $\boldsymbol{v}_{m+1}^{(k)}$ is of unit norm. Taking the norm of the error $\boldsymbol{\varepsilon}_m^{(k)}$ and inserting this yields

$$\begin{aligned}\|\boldsymbol{\varepsilon}_m^{(k)}\|_2 &\leq \|\boldsymbol{b}\|_2 \Big(\prod_{j=1}^{k} h_{m+1,m}^{(j)}\Big) \int_0^\infty \|\exp(-tA)\boldsymbol{v}_{m+1}^{(k)}\|_2 |\hat{f}^{(k+1)}(t)|\,\mathrm{d}t \\ &\leq \|\boldsymbol{b}\|_2 \Big(\prod_{j=1}^{k} h_{m+1,m}^{(j)}\Big) \mathcal{L}\{|\hat{f}^{(k+1)}|\}(\lambda_{\min}).\end{aligned}$$

We now replace $|\hat{f}^{(k+1)}|$ by the bound from Lemma 4.29 to obtain

$$\|\boldsymbol{\varepsilon}_m^{(k)}\|_2 \leq c\|\boldsymbol{b}\|_2 \Big(\prod_{j=1}^{k} h_{m+1,m}^{(j)} |\psi_m^{(j)}(-\omega)|\Big) \mathcal{L}\{\exp(t\omega)\}(\lambda_{\min}).$$

By hypothesis, we have $\lambda_{\min} > \omega$, so the Laplace transform $\mathcal{L}\{\exp(t\omega)\}(\lambda_{\min}) = (\lambda_{\min} - \omega)^{-1}$ exists, see Example 2.9. Moreover, $A - \omega I$ is Hermitian positive definite ($\lambda_{\min} - \omega > 0$), so we can use Lemma 4.30, which yields

$$\|\boldsymbol{\varepsilon}_m^{(k)}\|_2 \leq \|\boldsymbol{b}\|_2 c(\lambda_{\min} - \omega)^{-1} \sqrt{\lambda_{\max} - \omega}\; \|\boldsymbol{\epsilon}_m^{(k)}(-\omega)\|_{A-\omega I}.$$

Since $\sqrt{\kappa(A - \omega I)} = \sqrt{(\lambda_{\max} - \omega)(\lambda_{\min} - \omega)^{-1}}$, this proves the assertion. □

Any error bound for restarted CG can now be combined with Lemma 4.31 to show convergence. We use Theorem 2.75.

**Corollary 4.32.** *Under the assumptions of Lemma 4.31, the error satisfies*

$$\|\boldsymbol{\varepsilon}_m^{(k)}\|_2 \leq \|\boldsymbol{b}\|_2\, c\, \frac{\sqrt{\kappa(A - \omega I)}}{\lambda_{\min} - \omega} \gamma_m(-\omega)^k$$

*with the constants*

$$\gamma_m(-\omega) = \frac{1}{\cosh(m \log q(-\omega))} < 1, \quad q(-\omega) = \frac{\sqrt{\kappa(A - \omega I)} - 1}{\sqrt{\kappa(A - \omega I)} + 1}.$$

*Proof.* After inserting Theorem 2.75 into Lemma 4.31, we use

$$\|(A - \omega I)^{-1}\boldsymbol{b}\|_{A-\omega I} \leq \frac{\|\boldsymbol{b}\|_2}{\sqrt{\lambda_{\min} - \omega}},$$

see, e.g., [81, Eq. (5.12)]. □

**Remark 4.33.** Corollary 4.32 is similar to the error bound

$$\|\boldsymbol{\varepsilon}_m^{(k)}\|_2 \leq \|\boldsymbol{b}\|_2 \sqrt{\kappa(A)} f(\sqrt{\lambda_{\min}\lambda_{\max}}) \gamma_m(t_0)^k$$

for Stieltjes functions (Theorem 2.71) if both are applicable: Assume that Theorem 2.71 and Corollary 4.32 can both be applied, i.e., $A$ is Hermitian positive definite and $f(s) = \mathcal{L}\{\hat{f}\}(s)$ is not only a Laplace transform but can also be written as

$$f(s) = \int_0^\infty \frac{\rho(t)}{t+s}\,\mathrm{d}t$$

for $\rho \geq 0$. (In other words, $f$ is a Stieltjes function.) Then we have $\hat{f}(t) = \mathcal{L}\{\rho\}(t)$, so $\hat{f} \geq 0$ for $t \geq 0$. Moreover, since $\hat{f}(t)$ is a Laplace transform, we can choose $\omega = 0$ for an exponential bound, see [25, Theorem 23.7]. We insert this exponential bound, which yields

$$f(s) = \mathcal{L}\{|\hat{f}|\}(s) \leq c\mathcal{L}\{1\}(s) = cs^{-1}.$$

The term $f(\sqrt{\lambda_{\min}\lambda_{\max}})$ in Theorem 2.71 resolves to

$$f(\sqrt{\lambda_{\min}\lambda_{\max}}) \leq c\sqrt{\lambda_{\min}\lambda_{\max}}^{-1} \leq c\lambda_{\min}^{-1}.$$

We also have $\gamma_m(t_0) \leq \gamma_m(0)$ (see, e.g., [81, Proposition 5.6]). The bound in Theorem 2.71 now gives

$$\|\boldsymbol{\varepsilon}_m^{(k)}\|_2 \leq \|\boldsymbol{b}\|_2\, c\,\frac{\sqrt{\kappa(A)}}{\lambda_{\min}}\gamma_m(0)^k,$$

which is exactly Corollary 4.32 for our choice $\omega = 0$. Thus, our bound can be interpreted as a variation of Theorem 2.71 that applies to many Laplace transforms even if they are not Stieltjes functions. The disadvantage is that it is less tight.

### 4.4.3. A priori bound III: Main case

We now turn our attention to the case where $\hat{f}(t)$ can be exponentially bounded only for $t \geq T$ for some $T \geq 0$ and otherwise is integrable.

**Theorem 4.34.** *Let $\hat{f} \in L^1([0,T])$ and $|\hat{f}| \leq c\exp(t\omega)$ for almost all $t \geq T$. Let $A$ be Hermitian and its smallest eigenvalue $\lambda_{\min} > \omega$. The error of the restarted Arnoldi method for $\mathcal{L}\{\hat{f}\}(s)$ or for the complete Bernstein function with density $\mu'(t) = \hat{f}(t)$ in Definition 2.34 satisfies*

$$\frac{\|\boldsymbol{\varepsilon}_m^{(k)}\|_2}{\|\boldsymbol{b}\|_2} \leq C_0\frac{(C_1T)^{km}}{(km)!}\mathcal{L}\{|\hat{f}|\chi_{(0,T)}\}(\xi) + c\,\frac{\sqrt{\kappa(A-\omega I)}}{\lambda_{\min}-\omega}\gamma_m(-\omega)^k,$$

*where the constants $C_0$, $C_1$, $\xi$ are as in Lemma 4.25 and $\gamma_m(-\omega) < 1$ is defined in Corollary 4.32.*

*Proof.* As mentioned at the beginning of the section, we split the integral at $T$: We have

$$\begin{aligned}\|\boldsymbol{\varepsilon}_m^{(k)}\|_2 &= \|\int_0^\infty (\exp(-tA)\boldsymbol{b} - \boldsymbol{y}_m^{(k)}(t))\hat{f}(t)\,\mathrm{d}t\|_2 \\ &\leq \|\int_0^T (\exp(-tA)\boldsymbol{b} - \boldsymbol{y}_m^{(k)}(t))\hat{f}(t)\,\mathrm{d}t\|_2 + \|\int_T^\infty (\exp(-tA)\boldsymbol{b} - \boldsymbol{y}_m^{(k)}(t))\hat{f}(t)\,\mathrm{d}t\|_2\end{aligned}$$

where $\boldsymbol{y}_m^{(k)}$ is the restarted Arnoldi approximation to $\exp(-tA)\boldsymbol{b}$ as in Lemma 4.25. The first term is the error for $\mathcal{L}\{\hat{f}\chi_{(0,T)}\}(s)$, so we can apply Lemma 4.27. The second term is the error for $\mathcal{L}\{\hat{f}\chi_{(T,\infty)}\}(s)$. Since $|\hat{f}(t)\chi_{(T,\infty)}(t)| \leq c\exp(t\omega)$ for almost all $t \geq 0$, we can apply Corollary 4.32. Note that the same line of thought holds for complete Bernstein functions. □

For some complete Bernstein functions the property $\hat{f} \in L^1([0,T])$ is not satisfied. For the error representation in Lemma 4.10, we showed that it is sufficient to consider $\alpha\{t\hat{f}(t)\}$ instead of $\alpha\{\hat{f}(t)\}$. We can similarly modify the above error bound:

**Corollary 4.35.** *Let the conditions of Theorem 4.34 hold, but instead of $\hat{f} \in L^1([0,T])$, let $t\hat{f}(t) \in L^1([0,T])$. Then*

$$\frac{\|\boldsymbol{\varepsilon}_m^{(k)}\|_2}{\|\boldsymbol{b}\|_2} \leq C_0 \frac{C_1^{km} T^{km-1}}{(km)!} \mathcal{L}\{t|\hat{f}(t)|\chi_{(0,T)}(t)\}(\xi) + c\,\frac{\sqrt{\kappa(A-\omega I)}}{\lambda_{\min}-\omega}\gamma_m(-\omega)^k.$$

*Proof.* We changed the properties of $\hat{f}$ only in $[0,T]$, so only the first term in the proof of Theorem 4.34 is affected. The rest follows as in the proof for Lemma 4.27 with the difference that we use $t^{km} \leq tT^{km-1}$ instead of $t^{km} \leq T^{km}$. □

With Theorem 4.34 and Corollary 4.35, we have derived error bounds for many Laplace transforms and complete Bernstein functions. Besides the requirements on $A$, we only need an exponential bound for $|\hat{f}(t)|$ that is valid almost everywhere. In Section 4.5, we present such bounds for our previous examples from Section 4.3. This shall serve as an illustration that our error bound covers many relevant cases. There are, however, Laplace transforms and complete Bernstein functions that are not covered: Take, e.g.,

$$\hat{f}(t) = \sum_{j=0}^{\infty} j!\,\chi_{(j,\,j+(j!)^{-2})}(t).$$

Then it is easy to see that $|\hat{f}(t)|$ cannot be exponentially bounded but its Laplace transform exists with $\alpha\{\hat{f}\} \leq 0$. This is implied by

$$\mathcal{L}\{\hat{f}\}(0) = \sum_{j=0}^{\infty} j! \int_j^{j+(j!)^{-2}} \mathrm{d}t = \sum_{j=0}^{\infty} \frac{1}{j!} = \exp(1) < \infty,$$

where we interchanged the order of integration and (infinite) summation according to [77, Theorem 1.27] for the first equality.

**Remark 4.36.** We used the smallest eigenvalue $\lambda_{\min}$ of $A$ in the previous proofs. It is well-known that the Lanczos process captures an eigenvalue $\lambda$ only if the corresponding eigenvector $\boldsymbol{v}$ is not orthogonal to $\boldsymbol{b}$, i.e., if $\boldsymbol{v}^{\mathsf{H}}\boldsymbol{b} \neq 0$, see, e.g., [85] and cf. Remark 4.23. Accordingly, one can replace $\lambda_{\min}$ by a larger eigenvalue if it is known that $\boldsymbol{b}$ is orthogonal to the eigenvectors corresponding to smaller eigenvalues. One sometimes works

with this "effective" smallest eigenvalue $\lambda_{\min,\text{eff}}$ (and the resulting effective condition number) when analyzing the behavior of CG, see, e.g., [60, Eq. (8)]. It is easily verified that this can be incorporated into our bounds.

### 4.4.4. A posteriori bound

The a posteriori error bound for Stieltjes functions from [41], see Corollary 2.73, can be extended to Laplace transforms. Following the discussion in Section 4.1.2, it is easily modified for complete Bernstein functions. We state both cases using parentheses.

**Lemma 4.37.** *Let the assumptions of Corollary 4.8 (Lemma 4.10) hold. Let further $\hat{f}(t)$ be real and have constant sign for $t \geq 0$, $\alpha\{\hat{f}\} \leq 0$ ($\alpha\{t\hat{f}(t)\} \leq 0$) and $A$ be Hermitian positive definite. Define*

$$\widetilde{H}_m^{(k)} = \begin{bmatrix} H_m^{(k)} & h_{m+1,m}^{(k)} \boldsymbol{e}_m \\ h_{m+1,m}^{(k)} \boldsymbol{e}_m^{\mathsf{H}} & (h_{m+1,m}^{(k)})^2 \boldsymbol{e}_m^{\mathsf{T}} (H_m^{(k)} - aI)^{-1} \boldsymbol{e}_m \end{bmatrix}$$

*with $0 < a \leq \lambda_{\min}$. Let $f^{(k)}$ be the error function as in Algorithm 1, i.e.,*

$$\boldsymbol{\varepsilon}_m^{(k-1)} = \|\boldsymbol{b}\|_2 f^{(k)}(A) \boldsymbol{v}_{m+1}^{(k-1)},$$

*which means*

$$f^{(k)}(s) = (-1)^{k-1} \Big( \prod_{j=1}^{k-1} h_{m+1,m}^{(j)} \Big) \mathcal{L}\{\hat{f}^{(k)}\}(s).$$

*Then for $k \geq 2$,*

$$\|f^{(k)}(H_m^{(k)}) \boldsymbol{e}_1\|_2 \leq \frac{\|\boldsymbol{\varepsilon}_m^{(k-1)}\|_2}{\|\boldsymbol{b}\|_2} \leq \|f^{(k)}(\widetilde{H}_m^{(k)}) \boldsymbol{e}_1\|_2.$$

*Proof.* The proof is similar to the derivation of Corollary 2.73. Consider the norm of the error

$$\|\boldsymbol{\varepsilon}_m^{(k-1)}\|_2^2 = \|\boldsymbol{b}\|_2^2 (\boldsymbol{v}_{m+1}^{(k-1)})^{\mathsf{H}} \big(f^{(k)}(A)\big)^{\mathsf{H}} f^{(k)}(A) \boldsymbol{v}_{m+1}^{(k-1)}.$$

The matrix $A$ is Hermitian, so $\big(f^{(k)}(A)\big)^{\mathsf{H}} = f^{(k)}(A)$. This means we have the bilinear form

$$\frac{\|\boldsymbol{\varepsilon}_m^{(k-1)}\|_2^2}{\|\boldsymbol{b}\|_2^2} = (\boldsymbol{v}_{m+1}^{(k-1)})^{\mathsf{H}} (f^{(k)}(A))^2 \boldsymbol{v}_{m+1}^{(k-1)}$$

and similarly

$$\|f^{(k)}(H_m^{(k)}) \boldsymbol{e}_1\|_2^2 = \boldsymbol{e}_1^{\mathsf{T}} (f^{(k)}(H_m^{(k)}))^2 \boldsymbol{e}_1, \quad \|f^{(k)}(\widetilde{H}_m^{(k)}) \boldsymbol{e}_1\|_2^2 = \boldsymbol{e}_1^{\mathsf{T}} (f^{(k)}(\widetilde{H}_m^{(k)}))^2 \boldsymbol{e}_1.$$

If the function $(f^{(k)}(s))^2$ is a completely monotone function, then the lemma's assertion follows from Theorem 2.72. Note that

$$(f^{(k)}(s))^2 = \Big( \prod_{j=1}^{k-1} h_{m+1,m}^{(j)} \Big)^2 (\mathcal{L}\{\hat{f}^{(k)}\}(s))^2$$

is completely monotone if and only if $(\mathcal{L}\{\hat{f}^{(k)}\}(s))^2$ is completely monotone as the remaining term is a positive constant. Furthermore, the product of completely monotone functions is again completely monotone (Lemma 2.25), so it suffices to show that either $\mathcal{L}\{\hat{f}^{(k)}\}(s)$ or $-\mathcal{L}\{\hat{f}^{(k)}\}(s)$ is completely monotone. Since $\hat{f}^{(1)}$ has constant sign and $g^{(k)}$ has constant sign (Lemma 4.28), we know that $\hat{f}^{(k)}$ has constant sign, see the proof of Lemma 4.24. But this means that either $\hat{f}^{(k)}$ or $-\hat{f}^{(k)}$ is non-negative, which implies that $\pm\mathcal{L}\{\hat{f}^{(k)}\}(s)$ is completely monotone by Lemma 2.23. □

## 4.5. Numerical experiments II: Error bounds

We want to conclude this chapter by examining how the error bounds from Section 4.4 behave in practice. We apply them to the experiments from Section 4.3 with Hermitian test matrices. For our a priori error bound from Theorem 4.34, we need to investigate each function individually to obtain good (i.e., small) constants. The application of the a posteriori error bound from Lemma 4.37 is more straightforward. We only need to choose a value for $a$ such that $0 < a \leq \lambda_{\min}$, where $\lambda_{\min}$ is the smallest eigenvalue of $A$. In cycle $k$, we always choose the smallest eigenvalue of $H_m^{(j)}$, $j = 1, \ldots, k$, multiplied by a safety factor, which is set to $10^{-1}$ unless stated otherwise.

### 4.5.1. Fractional negative power less than $-1$

Consider again $\hat{f}(t) = 2\sqrt{\pi}^{-1}\sqrt{t}$ for the Laplace transform $f(s) = s^{-3/2} = \mathcal{L}\{\hat{f}\}(s)$ from Section 4.3.1. We can choose $T = 0$ and $\omega > 0$ but not $\omega = 0$. For a given $\omega$, we want to find the minimum value for $c$ such that $2\pi^{-1/2}\sqrt{t} \leq c\exp(t\omega)$. We start from

$$\frac{2}{\sqrt{\pi}}\sqrt{t} \leq c\exp(t\omega) \quad \Longrightarrow \quad c \geq \frac{2}{\sqrt{\pi}}\sqrt{t}\exp(-t\omega).$$

We look for the global maximum of the function on the very right as the inequality has to hold for every $t$. Because

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{2}{\sqrt{\pi}}\sqrt{t}\exp(-\omega t) = \frac{\exp(-\omega t)}{\sqrt{\pi t}}(1 - 2\omega t),$$

the maximum is achieved for $t = (2\omega)^{-1}$, from which we get

$$c \geq \sqrt{\frac{2}{\pi\omega\exp(1)}} \approx 0.484\,\omega^{-1/2}.$$

We apply Theorem 4.34[12] for several values of $\omega$ with the above $c$ for $A_{\mathrm{L}}$ with $N = 100$ in the left plot in Fig. 4.12. For $\omega \to 0$, we seem to approximately retrieve the observed convergence rate. The distance of the error bounds to the actual error is too large to be

[12] As we set $T = 0$, this is equivalent to using Corollary 4.32.

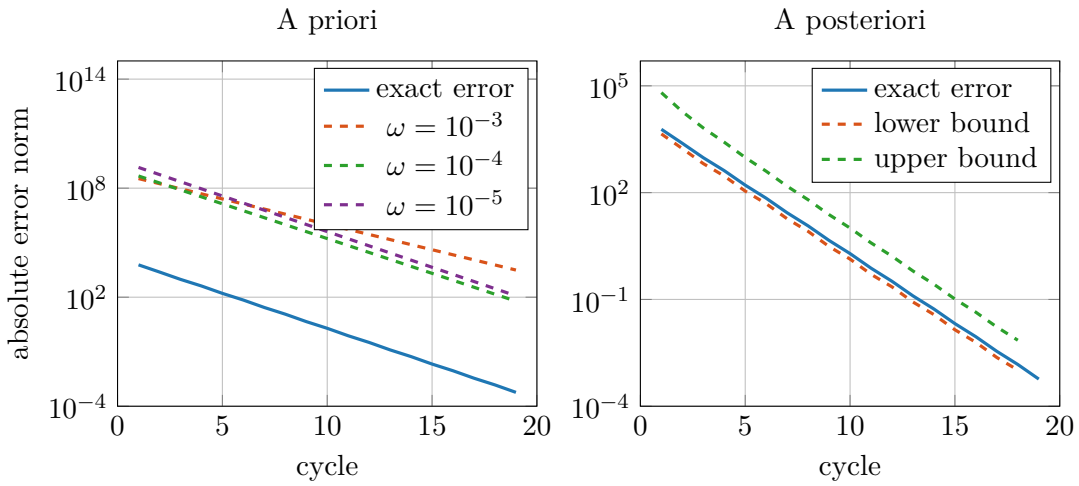


Figure 4.12.: Error norm and bounds for $A_{\mathrm{L}}^{-3/2}\boldsymbol{b}$ with $A_{\mathrm{L}}$ the 3D Laplace operator. Left: A priori by Theorem 4.34 with $c$ as small as possible for different values of $\omega$. Right: A posteriori by Lemma 4.37.

of any practical use, however. One might be tempted to change some of our parameter choices to obtain a tighter bound. Note that

$$\frac{\sqrt{\kappa(A_{\mathrm{L}})}}{\lambda_{\min}} \approx 10^{4.3},$$

which partly explains the large distance of the error bounds to the actual error. The rest comes from $c$; since $0 < \omega < \lambda_{\min} \approx 10^{-2.5}$, decreasing $c$ is only possible by choosing a value for $T$ larger than $(2\omega)^{-1}$. This, however, leads to an even worse error bound for a moderate number of matrix-vector products as it introduces the term $T^{km}$.

We next determine the a posteriori error bounds from Lemma 4.37 and plot them on the right in Fig. 4.12. We see that these error bounds are much tighter and give a good indication of the exact norm of the error.

### 4.5.2. Fractional diffusion processes on graphs

The function $\hat{f}(t) = (2\sqrt{\pi})^{-1}\exp(-(4t)^{-1})t^{-3/2}$ from Section 4.3.2 is integrable and converges to 0 for $t \to \infty$, so we can choose $T = 0$ and $\omega \geq 0$. Our matrices are graph Laplacians, which are always singular. Thus, $\lambda_{\min} > \omega$ cannot be fulfilled. As we use implicit desingularization (Remark 4.23), we can replace $\lambda_{\min}$ by its effective value $\lambda_{\min,\mathrm{eff}} > 0$ as mentioned in Remark 4.36. Accordingly, we choose $\omega = 0$.

Next, we need to determine $c$ such that

$$|\hat{f}(t)| = \frac{1}{2\sqrt{\pi}}\frac{\exp(-(4t)^{-1})}{t^{3/2}} \leq c\exp(0) = c.$$

We look again for the global maximum of $|\hat{f}|$ by differentiating. Using the product rule, this derivative follows as

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}|\hat{f}(t)| &= \frac{1}{4}\exp(-(4t)^{-1})t^{-7/2} - \frac{3}{2}\exp(-(4t)^{-1})t^{-5/2} \\ &= \frac{1}{4}\exp(-(4t)^{-1})t^{-5/2}(t^{-1}-6).\end{aligned}$$

The maximum occurs at $t = 6^{-1}$ giving

$$c \geq 3\sqrt{\frac{6}{\pi\exp(3)}} \approx 0.925.$$

To evaluate the a priori error bound from Theorem 4.34, we approximate the largest and the second smallest eigenvalue using MATLAB's `eigs()`. The resulting error bound is plotted in Fig. 4.13. We observe that it is again far from being practical due to several orders of magnitude in difference to the exact error norm. In addition, the slope of its curve differs significantly from the slope for the exact error norm, so the bound does not describe the observed rate of convergence very well.

The a posteriori error bounds are shown in Fig. 4.14. Here, we choose again the smallest eigenvalue of $H_m^{(k)}$ for $a$. We observed that with a safety factor of $10^{-1}$, the resulting estimate for the smallest eigenvalue of $A$ was larger than the exact one. We thus chose a safety factor of $10^{-3}$. Compared to the a priori bounds, the a posteriori bounds are much closer to the exact error norm and the curves of the error bounds show slopes similar to the curve of the error itself.

### 4.5.3. Gamma function

The gamma function from Section 4.3.3 needs two Laplace transforms since it is a two-sided Laplace transform. For the first one, we have $\hat{f}(t) = \exp(-\exp(-t))$, for which we can choose $T = 0$. It is easy to see that $0 < \hat{f}(t) < 1$ and that $\hat{f}(t)$ converges to 1 for $t \to \infty$. The smallest value for $\omega$ is thus $\omega = 0$ with $c \geq 1$.

The second transform $\mathcal{L}\{\hat{f}(-t)\}$ uses $\hat{f}(-t) = \exp(-\exp(t))$. This function converges to 0 faster than any exponential $\exp(\omega t)$, so we can choose any value for $\omega$. As finding an optimal value for $c$ is not trivial and we have $\omega = 0$ for the first transform $\mathcal{L}\{\hat{f}\}$, we choose again $\omega = 0$ and only need to find the maximum of $\exp(-\exp(t))$. This function decreases monotonically, so its largest value is attained at $t = 0$, which implies $c \geq \exp(-1)$. We combine the two bounds and obtain

$$\|\boldsymbol{\varepsilon}_m^{(k)}\|_2 \leq (1+\exp(-1))\|\boldsymbol{b}\|_2 \frac{\sqrt{\kappa(A_\mathrm{L})}}{\lambda_{\min}}\gamma_m(0)^k.$$

This bound together with the exact error norm is plotted in Fig. 4.15 on the left side. The right side shows the sum of the a posteriori error bounds from Lemma 4.37 for $\mathcal{L}\{\hat{f}(t)\}(A_\mathrm{L})$ and $\mathcal{L}\{\hat{f}(-t)\}(-A_\mathrm{L})$.

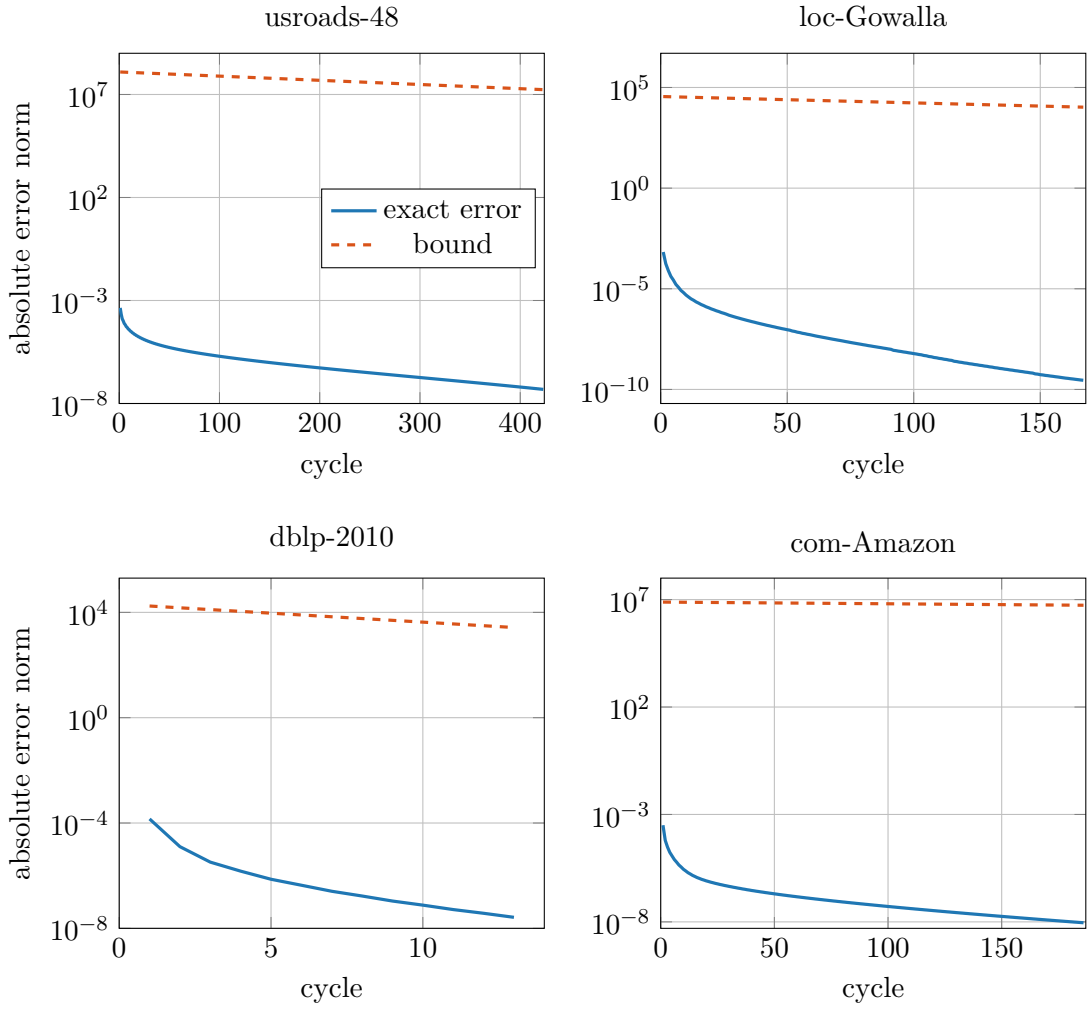


Figure 4.13.: Error norm and a priori bound (by Theorem 4.34) for $\exp(-\sqrt{L})\boldsymbol{b}$ with $L$ a graph Laplacian.

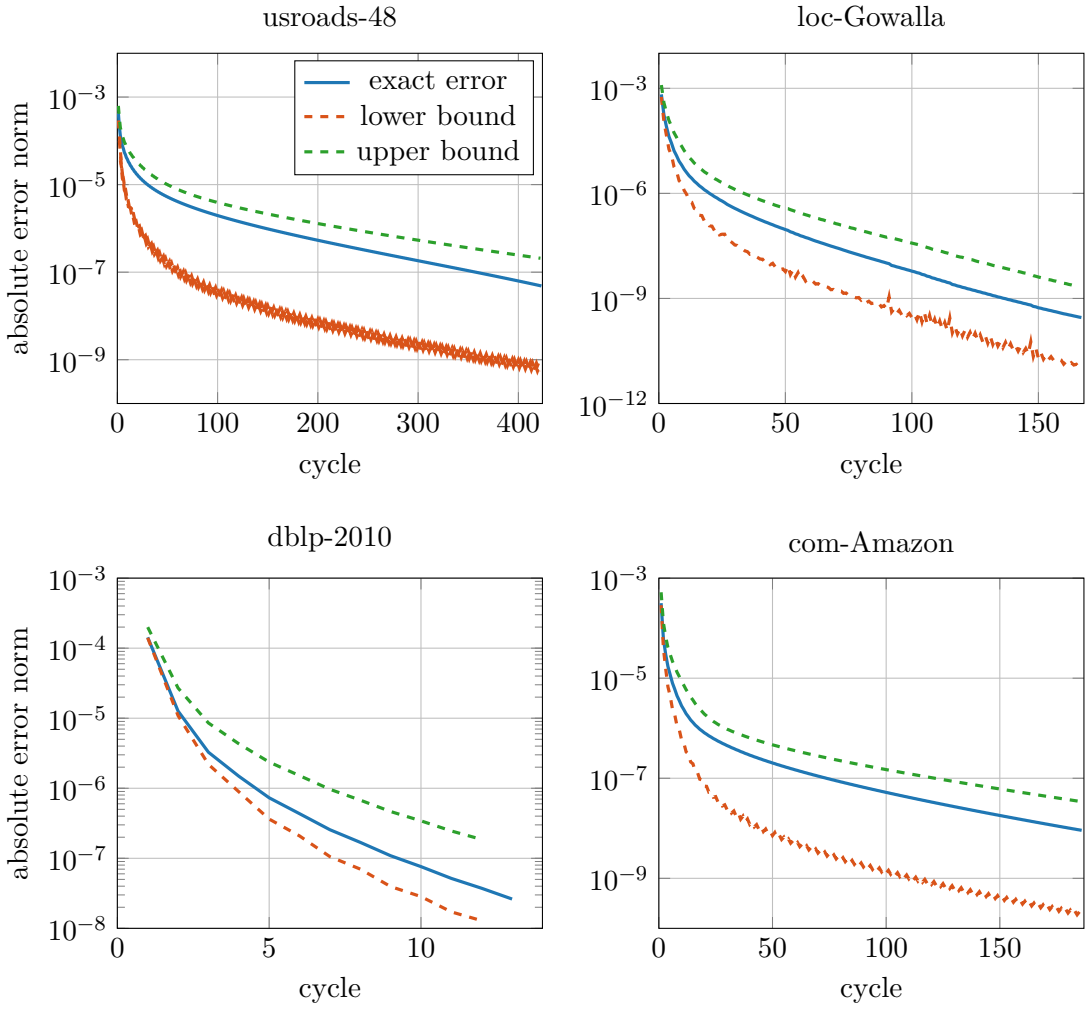


Figure 4.14.: Error norm and a posteriori bounds (by Lemma 4.37) for $\exp(-\sqrt{L})\boldsymbol{b}$ with $L$ a graph Laplacian.

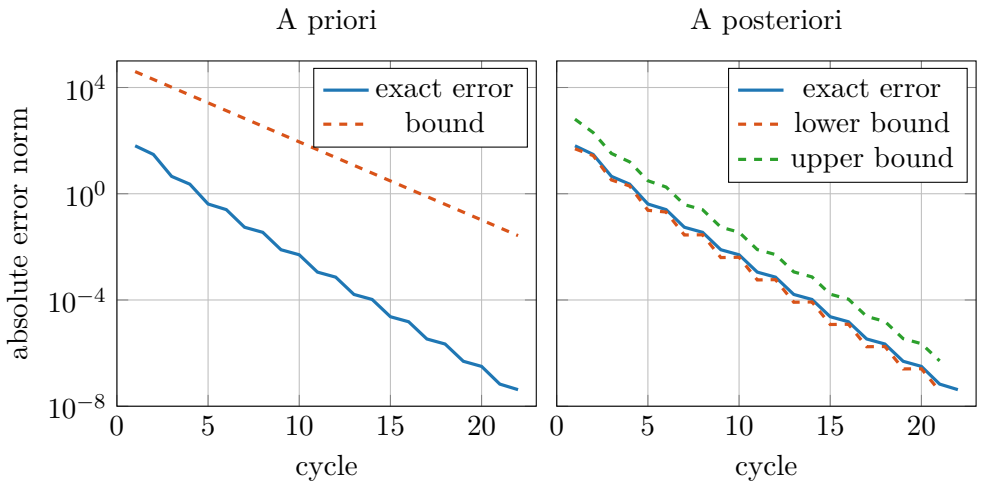


Figure 4.15.: Error norm and bounds for $\Gamma(A_\mathrm{L})\boldsymbol{b}$ with $A_\mathrm{L}$ the 2D Laplace operator. Left: A priori by Theorem 4.34 as explained in the running text. Right: A posteriori by Lemma 4.37.

### 4.5.4. Square root

Our first example of a complete Bernstein function was the square root in Section 4.3.4, for which we have the function $\hat{f}(t) = (2\sqrt{\pi})^{-1}t^{-3/2}$. Since $\hat{f}$ is not integrable, we need Corollary 4.35. We use the bound

$$|\hat{f}(t)| = \hat{f}(t) \leq \hat{f}(T)\exp(\omega t), \quad t > T > 0,\ \omega \geq 0,$$

and choose $\omega = 0$. For the 3D Laplace operator $A_\mathrm{L}$, we choose the contour $\Gamma$ such that it has a constant distance of 1 to $[0, 12] \supseteq \mathcal{W}(A_\mathrm{L})$ and the winding number is 1. Then the parameters of Lemma 4.25 that we need for Corollary 4.35 are

$$\ell(\Gamma) = 24 + 2\pi, \quad \operatorname{dist}(\Gamma, \mathcal{W}(A_\mathrm{L})) = 1, \quad \xi = -1.$$

The constants in Corollary 4.35 are thus

$$\begin{aligned} C_0 &= \frac{\ell(\Gamma)}{2\pi \operatorname{dist}(\Gamma, \mathcal{W}(A))} = \frac{24 + 2\pi}{2\pi} = \frac{12}{\pi} + 1, \\ C_1 &= \max_{s\in\Gamma} |s| + \rho(A_\mathrm{L}) = 13 + \rho(A_\mathrm{L}) \leq 25, \\ \mathcal{L}\{t\hat{f}(t)\chi_{(0,T)}(t)\}(\xi) &= \frac{1}{2\sqrt{\pi}} \int_0^T \exp(t)t^{-1/2}\,\mathrm{d}t \leq \frac{\exp(T)\sqrt{T}}{\sqrt{\pi}}. \end{aligned}$$

We summarize this as

$$\frac{\|\boldsymbol{\varepsilon}_m^{(k)}\|_2}{\|\boldsymbol{b}\|_2} \leq \left(\frac{12}{\pi} + 1\right) \frac{\exp(T)\sqrt{T}}{\sqrt{\pi}} \frac{25^{km}T^{km-1}}{(km)!} + \hat{f}(T)\frac{\sqrt{\kappa(A_\mathrm{L})}}{\lambda_{\min}}\gamma_m(0)^k.$$

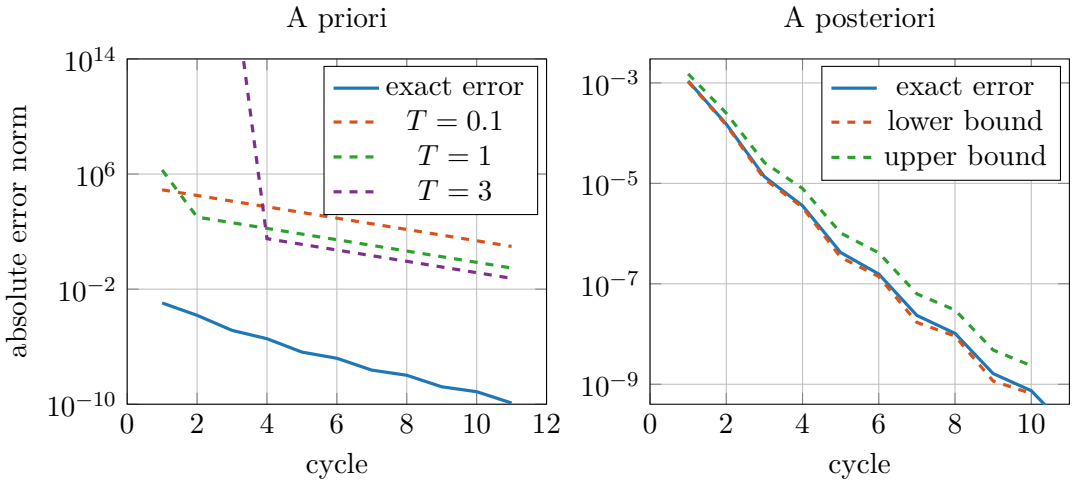


Figure 4.16.: Error norm and bounds for $A_{\mathrm{L}}^{1/2}\boldsymbol{b}$ with $A_{\mathrm{L}}$ the 3D Laplace operator. Left: A priori by Corollary 4.35 for different values of $T$ as explained in the running text. Right: A posteriori by Lemma 4.37.

We plot the resulting bound for several values of $T$ in Fig. 4.16 (left side). We also plot the a posteriori error bounds from Lemma 4.37 in the same figure on the right side. We see once again that the a priori error bound is far away from the exact error while the a posteriori bounds are very close. Moreover, we see that increasing $T$ reduces the distance of the a priori error bound to the exact error norm by a small amount if $k$ is large. For moderate $k$, however, the bound increases immensely when increasing $T$.

### 4.5.5. Entropy

Lastly, consider again the entropy from Section 4.3.5. We have $\hat{f}(t) = t^{-1}$, which is not integrable. We proceed similarly to Section 4.5.4: We choose again $\omega = 0$ (we need to use the effective smallest eigenvalue again) so that

$$|\hat{f}(t)| = \hat{f}(t) \leq T^{-1}\exp(0) = T^{-1}, \quad t > T > 0.$$

The contour is modified by replacing $[0, 12]$ by $[0, 1] \supseteq \mathcal{W}(A)$. This results in

$$C_0 = \frac{1}{\pi} + 1, \quad C_1 \leq 3, \quad \mathcal{L}\{t\hat{f}(t)\chi_{(0,T)}(t)\}(\xi) = \int_0^T \exp(t)\,\mathrm{d}t = \exp(T) - 1$$

and the bound

$$\frac{\|\boldsymbol{\varepsilon}_m^{(k)}\|_2}{\|\boldsymbol{b}\|_2} \leq \left(\frac{1}{\pi} + 1\right)(\exp(T) - 1)\frac{3^{km}T^{km-1}}{(km)!} + T^{-1}\frac{\sqrt{\kappa_{\mathrm{eff}}(A)}}{\lambda_{\min,\mathrm{eff}}}\gamma_m(0)^k.$$

The error bounds for three values of $T$ are plotted in Fig. 4.17. To better capture the asymptotic behavior, we increased the number of cycles to 10. The eigenvalues were

again approximated using `eigs()`. We observe that the bounds are even worse than for the previous examples. An explanation is again the smallest eigenvalue $\lambda_{\min,\mathrm{eff}}$. It is already close to 0 for the graph Laplacians $L$ and—as we obtained $A$ by dividing $L$ by its trace—we introduced the factor $\mathrm{trace}(L)$ into the bound via $\lambda_{\min,\mathrm{eff}}$. The trace is of order $10^6$ for our graphs, so we obtain a large constant in the linear term.

The a posteriori bounds from Lemma 4.37, on the other hand, are once again close to the exact error norm. We plot them in Fig. 4.18, where we again have chosen a safety factor of $10^{-3}$.

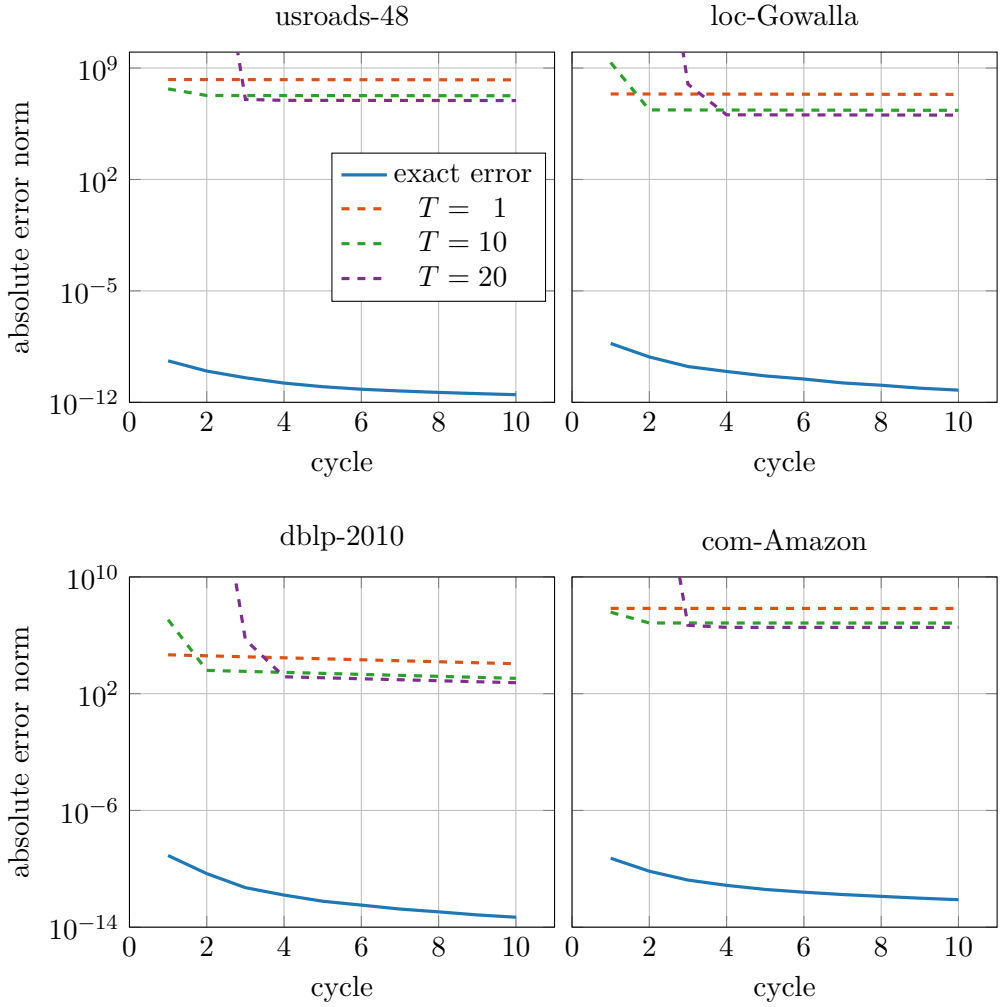


Figure 4.17.: Error norm and a priori bound (by Corollary 4.35) for $\log(A)\boldsymbol{b}$ with $A$ a normalized graph Laplacian.

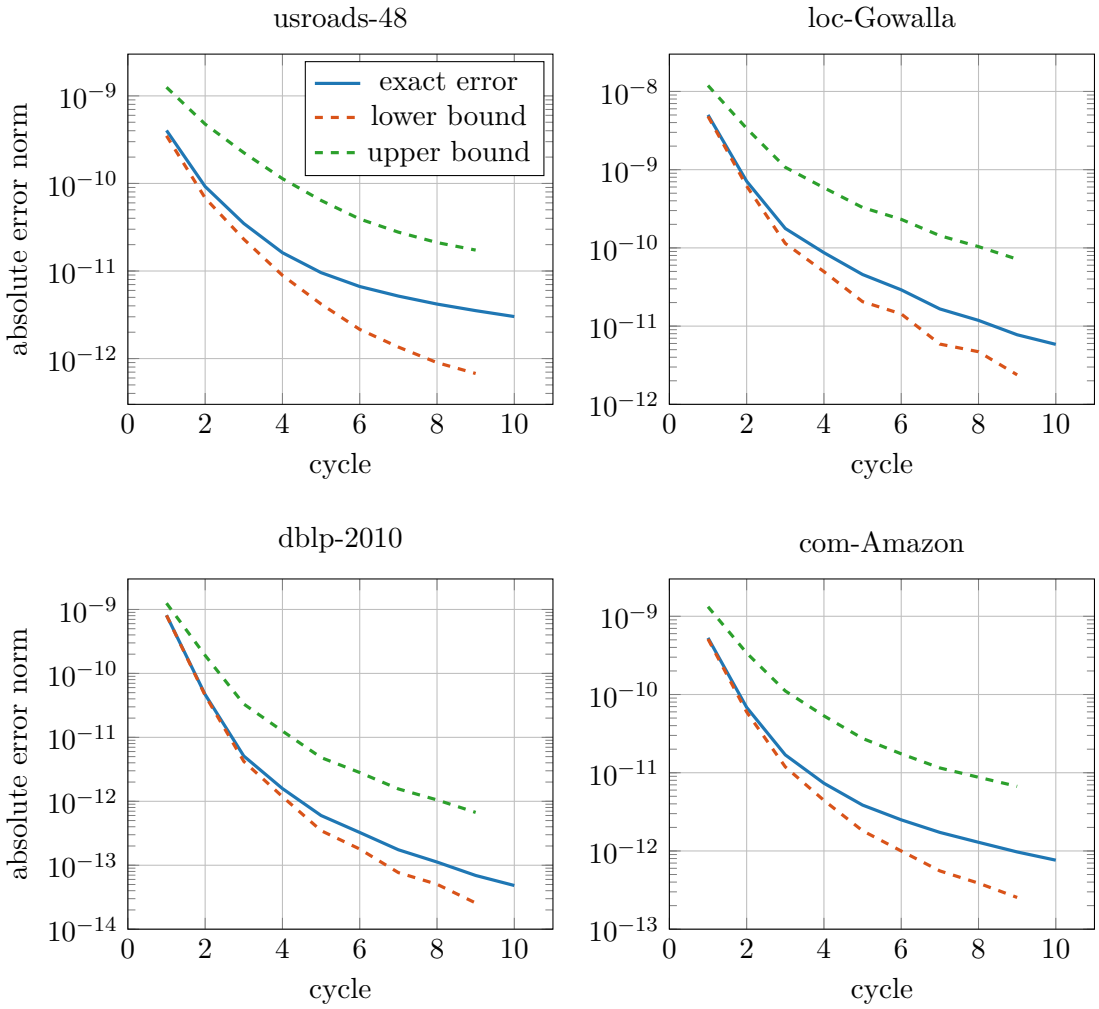


Figure 4.18.: Error norm and a posteriori bound (by Lemma 4.37) for $\log(A)\boldsymbol{b}$ with $A$ a normalized graph Laplacian.

# 5. Conclusions

We treated two topics regarding the computation of $f(A)\boldsymbol{b}$ in this thesis. First, we developed a new way of evaluating the action of rational matrix functions $r(A)\boldsymbol{b}$: In Section 3.1, we showed how to construct the CF-matrix $T_m(A)$ from a representation of the rational function $r(s)$ as a continued fraction. The computation of $r(A)\boldsymbol{b}$ essentially boils down to solving the linear system $T_m(A)\boldsymbol{x} = \boldsymbol{e}_1 \otimes \boldsymbol{b}$, where the CF-matrix $T_m(A)$ is block tridiagonal and each of its blocks has the same size as $A$.

We hoped that this reduction to a single linear system would lead to computationally faster methods than evaluating $r(A)\boldsymbol{b}$ using the shifted systems $(A - \tau_j I)^{-1}\boldsymbol{b}$ from the partial fraction expansion of $r(s)$. Since $T_m(A)$ is much larger than $A$, we focused our efforts on exploiting its block structure. With this in mind, we presented several theoretical results regarding Krylov subspace and multigrid methods in Section 3.2. They can be summarized as follows:

> *It seems one cannot beat the partial fraction expansion approach while exploiting the block structure of the CF-matrix.*

Because of that, we numerically investigated in Section 3.3 some ideas that do not exploit the structure. The CF-matrix approach did not offer any benefit there, either.

However, our approach might be beneficial in some cases, which we did not encounter here: There might be cases where a continued fraction representation of $r(s)$ is available but its partial fraction expansion is ill-conditioned. Then we can avoid calculating the expansion with the CF matrix approach and compute directly $r(A)\boldsymbol{b}$. Even if the partial fraction expansion is known, avoiding it might give a computational advantage: In some cases, the partial fraction expansion approach needs complex arithmetic, whereas our approach does not, see Remark 3.15. Then there are also fewer shifted systems from the partial fraction expansion, so this does not necessarily translate to a large computational advantage. Still, even a small speedup might be attractive. Lastly, our theoretical results focus on methods that use the Kronecker structure of $T_m(A)$. Despite our numerical experiments, some methods that do not exploit the Kronecker structure might yield a computational advantage when working with the CF-matrix. Future research could investigate these ideas in more detail.

We then turned our attention to the restarted Arnoldi method for $f(A)\boldsymbol{b}$ in Chapter 4. We developed in Section 4.1 a new representation of the error function if $f$ is a Laplace transform. As this representation describes the error as a new Laplace transform, it can be applied for further restarts, too. We showed that our result is easily extended to two-sided Laplace transforms and complete Bernstein functions. Using

this new representation, we described in Section 4.2 a possible implementation of the restarted Arnoldi method that is based on numerical integration. We suggested spline interpolation to avoid integrals defined recursively.

We implemented our ideas in MATLAB and subsequently illustrated in numerical experiments (Section 4.3) that efficient and stable restarts are possible for Laplace transforms and complete Bernstein functions: We tested our algorithm for several functions and various matrices including real-world graphs and were able to achieve our target accuracy in all cases. Thus, we have extended the class of functions for which the restarted Arnoldi method can be considered "black-box", i.e., for which a hand-tailored contour that depends on $A$ is not required. Some of these new functions could already be treated by the package `funm_quad` after some modifications like premultiplying the right-hand side $\boldsymbol{b}$ by $A$. In all our experiments, our algorithm needed up to a factor of 2 fewer matrix-vector products and can thus be considered faster. For Hermitian matrices, it even beat the two-pass Lanczos method in the number of required matrix-vector products in some cases. Indeed, for one function (the entropy in Section 4.3.5), the difference was again a factor of up to 2.

We then continued our theoretical investigation in Section 4.4: We developed an a priori error bound that is applicable to many Laplace transforms and complete Bernstein functions. The only requirement on $f$ is that the transformed function $\hat{f}$ grows at most exponentially. This bound proves that our algorithm converges at least linearly for these functions if $A$ is Hermitian and its smallest eigenvalue is not too small. Our numerical experiments in Section 4.5 showed that this bound is of minor practical relevance as it is several orders of magnitude larger than the exact error norm and often does not describe the observed rate of convergence very well. As a practical error bound, we extended an a posteriori error bound for Stieltjes functions to many Laplace transforms and complete Bernstein functions. We applied it to our experiments and observed that it is close to the exact error. As it also comes at a negligible cost, it can thus be used as a stopping criterion. Some care is required though as the upper a posteriori bound needs a good approximation of the smallest eigenvalue.

Future research could investigate how well other quadrature rules perform compared to the one of our implementation. The sinc quadrature (see Example 4.17), in particular, is a promising alternative as it has been used recently for $f(A)\boldsymbol{b}$ with the Laplace transforms $f(s) = s^{-\alpha}$, $\alpha \in (0, 1)$, see [20]. Moreover, while our a priori error bound covers many practically relevant functions, an a priori error bound that can be applied to *all* Laplace transforms and complete Bernstein functions might be interesting. A refinement of our bound such that it is closer to the exact error norm would also make it practically more relevant.

# A. Other definitions of the Laplace transform

In this appendix, we want to expand on Remark 2.10. In particular, we relate the different definitions of Laplace transforms and examine whether our error representation (Theorem 4.2 and Corollary 4.8) holds for those, too. As practically relevant functions are already included in the definition we used, we do not go too much into detail here.

## Improper Lebesgue integral

In [6, 25], Laplace transforms are defined as

$$\begin{aligned}\mathcal{L}^{\mathrm{im}}\{\hat{f}\}(s) \coloneqq \lim_{\omega\to\infty}\int_0^{\omega}\exp(-ts)\hat{f}(t)\,\mathrm{d}t &= \lim_{\omega\to\infty}\int_0^{\infty}\exp(-ts)\hat{f}(t)\chi_{(0,\omega)}(t)\,\mathrm{d}t\\ &= \lim_{\omega\to\infty}\mathcal{L}\{\hat{f}\chi_{(0,\omega)}\}(s),\end{aligned}$$

i.e., as an *improper* Lebesgue integral. Let us relate $\mathcal{L}^{\mathrm{im}}\{\hat{f}\}$ to our definition $\mathcal{L}\{\hat{f}\}$:

**Lemma A.1.** *Let one of the following be true:*

*(i)* $\mathcal{L}\{\hat{f}\}(s)$ *exists.*

*(ii)* $\mathcal{L}^{\mathrm{im}}\{\hat{f}\}(s)$ *converges absolutely.*

*Then the other statement is also true and*

$$\mathcal{L}\{\hat{f}\}(s) = \mathcal{L}^{\mathrm{im}}\{\hat{f}\}(s),$$

*i.e., the definition of [6, 25] coincides with Definition 2.1.*

*Proof.* By the Monotone Convergence Theorem (see [77, Theorem 1.26]), we have

$$\int_0^{\infty}\exp(-t\,\mathrm{Re}(s))|\hat{f}(t)|\,\mathrm{d}t = \lim_{\omega\to\infty}\int_0^{\infty}\exp(-t\,\mathrm{Re}(s))|\hat{f}(t)|\chi_{(0,\omega)}(t)\,\mathrm{d}t.$$

Hypothesis (i) says that the left side has a finite value whereas (ii) states that the right side has a finite value. This proves the first part. For the second part, note that the above implies that $\exp(-t\,\mathrm{Re}(s))|\hat{f}(t)|$ is integrable. As we also have

$$\exp(-t\,\mathrm{Re}(s))|\hat{f}(t)|\chi_{(0,\omega)}(t) \le \exp(-t\,\mathrm{Re}(s))|\hat{f}(t)|,$$

the Dominated Convergence Theorem (see [77, Theorem 1.34]) tells us that

$$\mathcal{L}^{\mathrm{im}}\{\hat{f}\}(s) = \int_0^\infty \exp(-ts)\hat{f}(t)(\lim_{\omega\to\infty} \chi_{(0,\omega)}(t))\,\mathrm{d}t = \mathcal{L}\{\hat{f}\}(s). \qquad \square$$

Note that for $\mathcal{L}^{\mathrm{im}}\{\hat{f}\}$, both regions of convergence (simple and absolute) are half-planes, see [25, Theorem 3.3, 3.5]. Lemma A.1 implies that as long as we restrict ourselves to the region of absolute convergence, everything we established in Chapter 4 still holds. It turns out that this restriction is not very large:

**Theorem A.2** (adapted from [25, Theorem 3.4])**.** *If $\mathcal{L}^{\mathrm{im}}\{\hat{f}\}(s_0)$ converges, then for any $s$ with $\mathrm{Re}(s) > \mathrm{Re}(s_0)$, we have*

$$\mathcal{L}^{\mathrm{im}}\{\hat{f}\}(s) = (s - s_0)\mathcal{L}\{\phi(t)\}(s - s_0),$$

*where*

$$\phi(t) = \int_0^t \exp(-s_0\tau)\hat{f}(\tau)\,\mathrm{d}\tau, \quad \alpha\{\phi\} \le 0.$$

This means that any improper Laplace transform $\mathcal{L}^{\mathrm{im}}\{\hat{f}\}(s)$ can be expressed by a proper Laplace transform $\mathcal{L}\{\phi\}(s - s_0)$ with an additional factor of $s - s_0$. The only exception is the case $s = s_0$. Choosing $s_0$ as small as possible shows us that we need to be in the interior of the region of convergence of $\mathcal{L}^{\mathrm{im}}\{\hat{f}\}$.

**Corollary A.3.** *Let*

$$\alpha^{\mathrm{im}}\{\hat{f}\} \coloneqq \inf\{\mathrm{Re}(s) : \mathcal{L}^{\mathrm{im}}\{\hat{f}\}(s) \text{ converges to a finite value}\}$$

*denote the abscissa of convergence of $\mathcal{L}^{\mathrm{im}}\{\hat{f}\}$. Let for some $s_0$*

$$\alpha^{\mathrm{im}}\{\hat{f}\} < \mathrm{Re}(s_0) < \min_{s\in\mathcal{W}(A)} \mathrm{Re}(s).$$

*Then*

$$\mathcal{L}^{\mathrm{im}}\{\hat{f}\}(A)\boldsymbol{b} = \mathcal{L}\{\phi\}(A - s_0 I)\left((A - s_0 I)\boldsymbol{b}\right).$$

*For this Laplace transform, we have*

$$\alpha\{\phi\} < \min_{s\in\mathcal{W}(A-s_0 I)} \mathrm{Re}(s),$$

*so our restart results Theorem 4.2 and Corollary 4.8 can be applied.*

## Other measures

Schilling et al. [80] use the proper integral

$$\mathcal{L}\{\mathrm{d}\mu\}(s) \coloneqq \int_0^\infty \exp(-ts)\,\mathrm{d}\mu(t) < \infty,$$

where $\mu$ is a positive measure.

**Lemma A.4.** *For any non-negative $\hat{f}$, there is a positive measure $\mu$ such that*

$$\mathcal{L}\{\hat{f}\}(s) = \mathcal{L}\{\mathrm{d}\mu\}(s)$$

*for every $s \in \mathbb{C}$. Conversely, for any positive $\sigma$-finite measure $\mu$ that is absolutely continuous with respect to the Lebesgue measure,[1] there is a non-negative function $\hat{f}$ such that the above equation is fulfilled for every $s \in \mathbb{C}$.*

*Proof.* The first statement follows immediately by defining

$$\mu(E) = \int_E \hat{f}(t)\,\mathrm{d}t.$$

The second statement is just the Radon-Nikodym Theorem, see [77, Theorem 6.10]. □

Note that $\hat{f}$ is non-negative in all our previous examples but there are Laplace transforms with oscillating $\hat{f}$, e.g., $\mathcal{L}\{\sin(t)\}(s) = (s^2+1)^{-1}$ ([25, Table of Laplace Transforms, No. 14]). On the other hand, it is easy to see that any function $\hat{f}$ that fulfills $\hat{f}(t)\,\mathrm{d}t = \mathrm{d}\mu(t)$ for positive $\mu(t)$ has to be non-negative. Thus, while the intersection of the two definitions is quite large, one is not contained in the other. It is not clear, however, which measures would yield interesting Laplace transforms but not allow a representation via a function $\hat{f}$. We give nonetheless a variation of our error representation from Corollary 4.8.

**Lemma A.5.** *Let $\mu$ be a positive $\sigma$-finite measure defined on the Borel $\sigma$-algebra. Denote by*

$$\alpha\{\mathrm{d}\mu(t)\} \coloneqq \inf\{\mathrm{Re}(s) : \mathcal{L}\{\mathrm{d}\mu(t)\}(s) \text{ exists}\}$$

*the abscissa of existence of $\mathcal{L}\{\mathrm{d}\mu(t)\}$. Let*

$$\max(\alpha\{\mathrm{d}\mu\}, 0) < \min_{s\in\mathcal{W}(A)} \mathrm{Re}(s).$$

*Then the error of the restarted Arnoldi method for $\mathcal{L}\{\mathrm{d}\mu\}(s)$ is*

$$\boldsymbol{\varepsilon}_m^{(k)} = \|\boldsymbol{b}\|_2(-1)^k\Big(\prod_{j=1}^{k} h_{m+1,m}^{(j)}\Big)\mathcal{L}\{\hat{f}^{(k+1)}\}(A)\boldsymbol{v}_{m+1}^{(k)}, \quad k \geq 1,$$

*with $\hat{f}^{(k+1)}$ defined as in Corollary 4.8 for $k \geq 2$ and*

$$\hat{f}^{(2)}(t) = \int_t^\infty g^{(1)}(\tau - t)\,\mathrm{d}\mu(t) \quad \text{for } k = 1.$$

[1] That is, if every null-set of the Lebesgue measure is a null-set of $\mu$.

*Proof.* The proof follows the ones for Theorem 4.2 (for $k = 1$) and Corollary 4.8 (for $k \geq 2$) if we replace $\hat{f}(t)\,\mathrm{d}t$ in the integrals by $\mathrm{d}\mu(t)$ (and equivalently $\mathcal{L}\{\hat{f}\}$ by $\mathcal{L}\{\mathrm{d}\mu(t)\}$). We mention the necessary tweaks:

The proof for $k = 1$ follows as for Theorem 4.2 with changes only when proving that the order of integration can be interchanged: First, one needs a more general form of Fubini's Theorem than Theorem 4.3, e.g., [77, Theorem 8.8] (for which we need that $\mu$ is $\sigma$-finite). By specifying that $\mu$ is defined on the Borel $\sigma$-algebra, the argument for Lemma 4.4 does not change, i.e., the integrand is measurable. The double integral of its absolute value is finite since $\mathcal{L}\{t\,\mathrm{d}\mu(t)\}(\nu)$ in Lemma 4.5 exists: Assuming that it exists, that transform is still the derivative of $\mathcal{L}\{\mathrm{d}\mu(t)\}$ by [80, Proof of Proposition 1.2]. The existence of the derivative is guaranteed by [80, Theorem 1.4] since $0 < \nu = \min_{s \in \mathcal{W}(A)} \mathrm{Re}(s)$.

For $k \geq 2$, note that the error for $k = 1$ is the Laplace transform of a function and not of a general measure anymore. Thus, we can use Corollary 4.8 to obtain the error representations for further restarts as long as $\alpha\{\hat{f}^{(2)}\} \leq \alpha\{\mathrm{d}\mu\}$. This follows exactly as the proof for Lemma 4.7 by replacing $\hat{f}(t)\,\mathrm{d}t$ by $\mathrm{d}\mu(t)$. □

# Bibliography

## Own publications


[L] A. Frommer, K. Kahl, M. Schweitzer, and M. Tsolakis, *Krylov subspace restarting for matrix Laplace transforms*, SIAM J. Matrix Anal. Appl., 44 (2023), pp. 693–717, doi: 10.1137/22M1499674.

[CF] A. Frommer, K. Kahl, and M. Tsolakis, *Matrix functions via linear systems built from continued fractions*, arXiv preprint arXiv:2109.03527 (2021), doi: 10.48550/ARXIV.2109.03527.


## Other publications


[1] M. Afanasjew, M. Eiermann, O. G. Ernst, and S. Güttel, *Implementation of a restarted Krylov subspace method for the evaluation of matrix functions*, Linear Algebra Appl., 429 (2008), pp. 2293–2314, doi: 10.1016/j.laa.2008.06.029.

[2] M. I. Ahmad, D. B. Szyld, and M. B. van Gijzen, *Preconditioned multishift BiCG for $\mathcal{H}_2$-optimal model reduction*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 401–424, doi: 10.1137/130914905.

[3] R. AlAhmad, *Laplace transform of the product of two functions*, Italian J. Pure Appl. Math, 44 (2020), pp. 800–804, url: https://ijpam.uniud.it/online_issue/202044/71%20RamiAlahmad.pdf.

[4] A. H. Al-Mohy and N. J. Higham, *A new scaling and squaring algorithm for the matrix exponential*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 970–989, doi: 10.1137/09074721X.

[5] A. H. Al-Mohy and N. J. Higham, *Computing the action of the matrix exponential, with an application to exponential integrators*, SIAM J. Sci. Comput., 33 (2011), pp. 488–511, doi: 10.1137/100788860.

[6] W. Arendt, C. J. K. Batty, M. Hieber, and F. Neubrander, *Vector-valued Laplace Transforms and Cauchy Problems*, 2nd ed., Springer, Basel, Switzerland, 2011, doi: 10.1007/978-3-0348-0087-7.

[7] M. Arioli and D. Loghin, *Discrete interpolation norms with applications*, SIAM J. Numer. Anal., 47 (2009), pp. 2924–2951, doi: 10.1137/080729360.

[8] R. K. Beatson, *On the convergence of some cubic spline interpolation schemes*, SIAM J. Numer. Anal., 23 (1986), pp. 903–912, doi: 10.1137/0723058.

[9] N. Bell, L. N. Olson, and J. Schroder, *PyAMG: Algebraic multigrid solvers in Python*, J. Open Source Softw., 7 (2022), p. 4142, doi: 10.21105/joss.04142.

[10] M. Benzi, M. Rinelli, and I. Simunec, *Computation of the von Neumann entropy of large matrices via trace estimators and rational Krylov methods*, arXiv preprint arXiv:2212.09642 (2022), doi: 10.48550/ARXIV.2212.09642.

[11] M. Benzi and I. Simunec, *Rational Krylov methods for fractional diffusion problems on graphs*, BIT, 62 (2022), pp. 357–385, doi: 10.1007/s10543-021-00881-0.

[12] C. Berg, *Stieltjes-Pick-Bernstein-Schoenberg and their connection to complete monotonicity*, in: *Positive Definite Functions. From Schoenberg to Space-Time Challenges*, ed. by J. Mateu and E. Porcu, University Jaume I, Castellón de la Plana, Spain, 2008.

[13] C. Berg and G. Forst, *Potential Theory on Locally Compact Abelian Groups*, Springer, Berlin, Germany, 1975.

[14] M. Berljafa and S. Güttel, *The RKFIT algorithm for nonlinear rational approximation*, SIAM J. Sci. Comput., 39 (2017), A2049–A2071, doi: 10.1137/15M1025426.

[15] R. P. Boas, *Entire Functions*, Academic Press, New York, USA, 1954.

[16] C. de Boor, *A Practical Guide to Splines*, revised ed., Springer, New York, USA, 2001.

[17] A. Boriçi, A. D. Kennedy, B. J. Pendleton, and U. Wenger, *The overlap operator as a continued fraction*, Nucl. Phys. B - Proceedings Supplements, 106 (2002), pp. 757–759, doi: 10.1016/S0920-5632(01)01835-7.

[18] J. Brannick, F. Cao, K. Kahl, R. D. Falgout, and X. Hu, *Optimal interpolation and compatible relaxation in classical algebraic multigrid*, SIAM J. Sci. Comput., 40 (2018), A1473–A1493, doi: 10.1137/17M1123456.

[19] J. R. Cardoso and A. Sadeghi, *Computation of matrix gamma function*, BIT, 59 (2019), pp. 343–370, doi: 10.1007/s10543-018-00744-1.

[20] A. A. Casulli and L. Robol, *Low-rank tensor structure preservation in fractional operators by means of exponential sums*, arXiv preprint arXiv:2208.05189 (2022), doi: 10.48550/ARXIV.2208.05189.

[21] M. Crouzeix and C. Palencia, *The numerical range is a* $(1+\sqrt{2})$*-spectral set*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 649–655, doi: 10.1137/17M1116672.

[22] A. Cuyt, V. B. Petersen, B. Verdonk, H. Waadeland, and W. B. Jones, *Handbook of Continued Fractions for Special Functions*, Springer, Dordrecht, Netherlands, 2008, doi: 10.1007/978-1-4020-6949-9.

[23] P. I. Davies and N. J. Higham, *Computing $f(A)b$ for Matrix Functions $f$*, in: *QCD and Numerical Analysis III*, ed. by A. Boriçi, A. Frommer, B. Joó, A. D. Kennedy, and B. Pendleton, Springer, Berlin, Germany, 2005, pp. 15–24, doi: 10.1007/3-540-28504-0_2.

[24] T. A. Davis and Y. Hu, *The University of Florida Sparse Matrix Collection*, ACM Trans. Math. Software, 38 (2011), pp. 1–25, doi: 10.1145/2049662.2049663.

[25] G. Doetsch, *Introduction to the Theory and Application of the Laplace Transformation*, Springer, Berlin, Germany, 1974, doi: 10.1007/978-3-642-65690-3.

[26] V. Druskin, *On monotonicity of the Lanczos approximation to the matrix exponential*, Linear Algebra Appl., 429 (2008), pp. 1679–1683, doi: 10.1016/j.laa.2008.04.046.

[27] V. Druskin, A. Greenbaum, and L. Knizhnerman, *Using nonorthogonal Lanczos vectors in the computation of matrix functions*, SIAM J. Sci. Comput., 19 (1998), pp. 38–54, doi: 10.1137/S1064827596303661.

[28] V. Druskin and L. Knizhnerman, *Extended Krylov subspaces: Approximation of the matrix square root and related functions*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 755–771, doi: 10.1137/S0895479895292400.

[29] M. Eiermann, O. G. Ernst, and S. Güttel, *Deflated restarting for matrix functions*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 621–641, doi: 10.1137/090774665.

[30] M. Eiermann and O. G. Ernst, *A restarted Krylov subspace method for the evaluation of matrix functions*, SIAM J. Numer. Anal., 44 (2006), pp. 2481–2504, doi: 10.1137/050633846.

[31] M. Embree, *The tortoise and the hare restart GMRES*, SIAM Rev., 45 (2003), pp. 259–266, doi: 10.1137/S003614450139961.

[32] J. van den Eshof, A. Frommer, T. Lippert, K. Schilling, and H. A. van der Vorst, *Numerical methods for the QCD overlap operator. I. Sign-function and error bounds*, Comput. Phys. Commun., 146 (2002), pp. 203–224, doi: 10.1016/S0010-4655(02)00455-1.

[33] V. Faber, J. Liesen, and P. Tichý, *On the Forsythe conjecture*, arXiv preprint arXiv:2209.14579 (2022), doi: 10.48550/ARXIV.2209.14579.

[34] V. Faber and T. Manteuffel, *Necessary and sufficient conditions for the existence of a Conjugate Gradient method*, SIAM J. Numer. Anal., 21 (1984), pp. 352–362, doi: 10.1137/0721026.

[35] K. V. Fernando, *On computing an eigenvector of a tridiagonal matrix. Part I: Basic results*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 1013–1034, doi: 10.1137/S0895479895294484.

[36] G. E. Forsythe, *On the asymptotic directions of the s-dimensional optimum gradient method*, Numer. Math., 11 (1968), pp. 57–76, doi: 10.1007/BF02165472.

[37] M. Freund and E. Görlich, *Polynomial approximation of an entire function and rate of growth of Taylor coefficients*, Proc. Edinb. Math. Soc., 28 (1985), pp. 341–348, doi: 10.1017/S0013091500017156.

[38] A. Frommer and U. Gläßner, *Restarted GMRES for shifted linear systems*, SIAM J. Sci. Comput., 19 (1998), pp. 15–26, doi: 10.1137/S1064827596304563.

[39] A. Frommer, S. Güttel, and M. Schweitzer, *Convergence of restarted Krylov subspace methods for Stieltjes functions of matrices*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 1602–1624, doi: 10.1137/140973463.

[40] A. Frommer, S. Güttel, and M. Schweitzer, *Efficient and stable Arnoldi restarts for matrix functions based on quadrature*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 661–683, doi: 10.1137/13093491X.

[41] A. Frommer and M. Schweitzer, *Error bounds and estimates for Krylov subspace approximations of Stieltjes matrix functions*, BIT, 56 (2016), pp. 865–892, doi: 10.1007/s10543-015-0596-3.

[42] A. Frommer, S. Güttel, and M. Schweitzer, *FUNM_QUAD: An implementation of a stable, quadrature-based restarted Arnoldi method for matrix functions*, tech. rep., Bergische Universität Wuppertal, Germany, 2014.

[43] W. Gautschi, *The condition of polynomials in power form*, Math. Comp., 33 (1979), pp. 343–352, doi: 10.2307/2006047.

[44] G. H. Golub and C. F. van Loan, *Matrix Computations*, 4th ed., Johns Hopkins University Press, Baltimore, USA, 2013.

[45] G. H. Golub and G. Meurant, *Matrices, moments and quadrature*, in: *Numerical Analysis 1993*, ed. by D. F. Griffiths and G. A. Watson, Longman Scientific & Technical, Essex, United Kingdom, 1994, pp. 105–156.

[46] G. H. Golub and G. Meurant, *Matrices, Moments and Quadrature with Applications*, Princeton University Press, Princeton, USA, 2010.

[47] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM, Philadelphia, USA, 1997.

[48] S. Güttel, *Rational Krylov approximation of matrix functions: Numerical methods and optimal pole selection*, GAMM-Mitt., 36 (2013), pp. 8–31, doi: 10.1002/gamm.201310002.

[49] S. Güttel and M. Schweitzer, *A comparison of limited-memory Krylov methods for Stieltjes functions of Hermitian matrices*, SIAM J. Matrix Anal. Appl., 42 (2021), pp. 83–107, doi: 10.1137/20M1351072.

[50] S. Güttel and M. Schweitzer, *Randomized sketching for Krylov approximations of large-scale matrix functions*, arXiv preprint arXiv:2208.11447 (2022), doi: 10.48550/ARXIV.2208.11447.

[51] W. Hackbusch, *Hierarchical Matrices: Algorithms and Analysis*, Springer, Berlin, Germany, 2015, doi: 10.1007/978-3-662-47324-5.

[52] N. Hale, N. J. Higham, and L. N. Trefethen, *Computing $A^\alpha, \log(A)$, and related matrix functions by contour integrals*, SIAM J. Numer. Anal., 46 (2008), pp. 2505–2523, doi: 10.1137/070700607.

[53] P. Henrici, *Applied and Computational Complex Analysis, Vol. 1*, John Wiley & Sons, New York, USA, 1974.

[54] P. Henrici, *Applied and Computational Complex Analysis, Vol. 2*, John Wiley & Sons, New York, USA, 1977.

[55] N. J. Higham, *The scaling and squaring method for the matrix exponential revisited*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 1179–1193, doi: 10.1137/04061101X.

[56] N. J. Higham, *Functions of Matrices: Theory and Computation*, SIAM, Philadelphia, USA, 2008, doi: 10.1137/1.9780898717778.

[57] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, United Kingdom, 1991, doi: 10.1017/CBO9780511840371.

[58] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed., Cambridge University Press, Cambridge, United Kingdom, 2013.

[59] M. Ilić, I. W. Turner, and D. P. Simpson, *A restarted Lanczos approximation to functions of a symmetric matrix*, IMA J. Numer. Anal., 30 (2010), pp. 1044–1061, doi: 10.1093/imanum/drp003.

[60] K. Kahl and H. Rittich, *The deflated conjugate gradient method: Convergence, perturbation and accuracy*, Linear Algebra Appl., 515 (2017), pp. 111–129, doi: 10.1016/j.laa.2016.10.027.

[61] L. Knizhnerman and V. Simoncini, *A new investigation of the extended Krylov subspace method for matrix function evaluations*, Numer. Linear Algebra Appl., 17 (2010), pp. 615–638, doi: 10.1002/nla.652.

[62] A. Knopfmacher and J. Knopfmacher, *Maximum Length of the Euclidean Algorithm and Continued Fractions in F(X)*, in: *Applications of Fibonacci Numbers*, ed. by G. E. Bergum, A. N. Philippou, and A. F. Horadam, Springer, Dordrecht, Netherlands, 1990, pp. 217–222, doi: 10.1007/978-94-009-1910-5_25.

[63] A. B. J. Kuijlaars, *Which eigenvalues are found by the Lanczos method?*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 306–321, doi: 10.1137/S089547989935527X.

[64] J. Liesen and Z. Strakoš, *Krylov Subspace Methods*, Oxford University Press, London, United Kingdom, 2013.

[65] J. Liesen and P. Tichý, *Convergence analysis of Krylov subspace methods*, GAMM-Mitt., 27 (2004), pp. 153–173, doi: 10.1002/gamm.201490008.

[66] J. Mahoney and B. Sivazlian, *Partial fractions expansion: a review of computational methodology and efficiency*, J. Comput. Appl. Math., 9 (1983), pp. 247–269, doi: 10.1016/0377-0427(83)90018-3.

[67] P.-G. Martinsson and J. A. Tropp, *Randomized numerical linear algebra: Foundations and algorithms*, Acta Numer., 29 (2020), pp. 403–572, doi: 10.1017/S0962492920000021.

[68] MATLAB, *version 9.10.0 (R2021a)*, The MathWorks Inc., Natick, USA, 2022.

[69] A. McCurdy, K. C. Ng, and B. N. Parlett, *Accurate computation of divided differences of the exponential function*, Math. Comp., 43 (1984), pp. 501–528, doi: 10.2307/2008291.

[70] G. Meurant and Z. Strakoš, *The Lanczos and conjugate gradient algorithms in finite precision arithmetic*, Acta Numer., 15 (2006), pp. 471–542, doi: 10.1017/S096249290626001X.

[71] J. Miklosko, *Investigation of algorithms for numerical computation of continued fractions*, USSR Comput. Math. Math. Phys., 16 (1976), pp. 1–12, doi: 10.1016/0041-5553(76)90001-X.

[72] S. Miyajima, *Verified computation of matrix gamma function*, Linear Multilinear Algebra, 70 (2022), pp. 1207–1229, doi: 10.1080/03081087.2020.1757602.

[73] C. Moler and C. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Rev., 45 (2003), pp. 3–49, doi: 10.1137/S00361445024180.

[74] Y. Nakatsukasa and J. A. Tropp, *Fast & accurate randomized algorithms for linear systems and eigenvalue problems*, arXiv preprint arXiv:2111.00113 (2021), doi: 10.48550/ARXIV.2111.00113.

[75] H. Neuberger, *Overlap lattice Dirac operator and dynamical fermions*, Phys. Rev. D, 60 (1999), p. 065006, doi: 10.1103/PhysRevD.60.065006.

[76] G. Pleiss, M. Jankowiak, D. Eriksson, A. Damle, and J. Gardner, *Fast matrix square roots with applications to Gaussian processes and Bayesian optimization*, in: *Adv. Neural Inf. Process. Syst.* Vol. 33, 2020, pp. 22268–22281.

[77] W. Rudin, *Real and Complex Analysis*, 3rd ed., McGraw-Hill, New York, USA, 1987.

[78] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, USA, 2003, doi: 10.1137/1.9780898718003.

[79] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, 2nd ed., SIAM, Philadelphia, USA, 2011, doi: 10.1137/1.9781611970739.

[80] R. L. Schilling, R. Song, and Z. Vondraček, *Bernstein Functions*, 2nd ed., De Gruyter, Berlin, Germany, 2012, doi: 10.1515/9783110269338.

[81] M. Schweitzer, *Restarting and error estimation in polynomial and extended Krylov subspace methods for the approximation of matrix functions*, Doctorate Thesis, Bergische Universität Wuppertal, Germany, 2015.

[82] L. F. Shampine, *Vectorized adaptive quadrature in MATLAB*, J. Comput. Appl. Math., 211 (2008), pp. 131–140, doi: 10.1016/j.cam.2006.11.021.

[83] V. Simoncini, *Computational methods for linear matrix equations*, SIAM Rev., 58 (2016), pp. 377–441, doi: 10.1137/130912839.

[84] V. Simoncini and D. B. Szyld, *Recent computational developments in Krylov subspace methods for linear systems*, Numer. Linear Algebra Appl., 14 (2007), pp. 1–59, doi: 10.1002/nla.499.

[85] A. van der Sluis and H. A. van der Vorst, *The rate of convergence of Conjugate Gradients*, Numer. Math., 48 (1986), pp. 543–560, doi: 10.1007/BF01389450.

[86] F. Stenger, *Numerical Methods Based on Sinc and Analytic Functions*, Springer, New York, USA, 1993, doi: 10.1007/978-1-4612-2706-9.

[87] G. W. Stewart and J. Sun, *Matrix Perturbation Theory*, Academic Press, Boston, USA, 1990.

[88] H. Tal-Ezer, *On restart and error estimation for Krylov approximation of* $w = f(A)v$, SIAM J. Sci. Comput., 29 (2007), pp. 2426–2441, doi: 10.1137/040617868.

[89] L. N. Trefethen, *Approximation Theory and Approximation Practice*, SIAM, Philadelphia, USA, 2013.

[90] U. Trottenberg, C. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, San Diego, USA, 2000.

[91] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C J Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, *SciPy 1.0: Fundamental algorithms for scientific computing in python*, Nat. Methods, 17 (2020), pp. 261–272, doi: 10.1038/s41592-019-0686-2.

[92] H. Wang and Q. Ye, *Error bounds for the Krylov subspace methods for computations of matrix exponentials*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 155–187, doi: 10.1137/16M1063733.

[93] J. A. C. Weidemann and B. Fornberg, *Fully numerical Laplace transform methods*, Numer. Algorithms, 92 (2023), pp. 985–1006, doi: 10.1007/s11075-022-01368-x.

[94] D. V. Widder, *The Laplace Transform*, Princeton University Press, Princeton, USA, 1952.

[95] J. Xu and L. Zikatanov, *Algebraic multigrid methods*, Acta Numer., 26 (2017), pp. 591–721, doi: 10.1017/S0962492917000083.

[96] D. M. Young, *Iterative Solution of Large Linear Systems*, Academic Press, Orlando, USA, 1971, doi: 10.1016/C2013-0-11733-3.