Topics in the computation of matrix functions

Fréchet derivatives, low-rank updates, decay bounds and limited-memory algorithms

Kumulative Habilitationsschrift

Marcel Schweitzer

Bergische Universität Wuppertal
Fakultät für Mathematik und Naturwissenschaften
Wuppertal, 10.05.2024

ACKNOWLEDGMENTS

Writing this thesis would not have been possible without the help and support of many people.

I wish to thank all my collaborators and friends in the numerical linear algebra community with whom I discussed about and worked on what ended up becoming the publications that are summarized in this thesis.

Additionally, I'd like to thank my colleagues in the "Scientific Computing and High Performance Computing" group at the University of Wuppertal who have created a great working environment and made writing this thesis (as well as many other things) a very enjoyable endeavor.

Of course I also want to thank my mentor Andreas Frommer, not only for always encouraging me in my research, but also for being an extremely nice person, and for initially sparking my interest in numerical linear algebra when I was still an undergraduate student almost exactly 15 years ago.

Last, but definitely not least, I would like to thank my family and friends for all the love, support and good times we had. In particular, I am infinitely grateful to Sonja, who always unconditionally supports me in everything I do. There are no words that could express how important you are to me.

COLLECTION OF PUBLICATIONS SUMMARIZED IN THIS THESIS

[S1] B. Arslan, S. D. Relton, and M. Schweitzer, *Structured level-2 condition numbers of matrix functions*, Electron. J. Linear Algebra, 40 (2024), pp. 28–47.

[S2] B. Beckermann, A. Cortinovis, D. Kressner, and M. Schweitzer, *Low-rank updates of matrix functions II: Rational Krylov methods*, SIAM J. Numer. Anal., 59 (2021), pp. 1325–1347.

[S3] B. Beckermann, D. Kressner, and M. Schweitzer, *Low-rank updates of matrix functions*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 539–565.

[S4] M. A. Botchev, L. A. Knizhnerman, and M. Schweitzer, *Krylov subspace residual and restarting for certain second order differential equations*, SIAM J. Sci. Comput., 46 (2024), pp. S223–S253.

[S5] 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.

[S6] A. Frommer, C. Schimmel, and M. Schweitzer, *Bounds for the decay of the entries in inverses and Cauchy–Stieltjes functions of certain sparse, normal matrices*, Numer. Linear Algebra Appl., 25 (2018), p. e2131.

[S7] A. Frommer, C. Schimmel, and M. Schweitzer, *Non-Toeplitz decay bounds for inverses of Hermitian positive definite tridiagonal matrices*, Electron. Trans. Numer. Anal., 48 (2018), pp. 362–372.

[S8] A. Frommer, C. Schimmel, and M. Schweitzer, *Analysis of probing techniques for sparse approximation and trace estimation of decaying matrix functions*, SIAM J. Matrix Anal. Appl., 42 (2021), pp. 1290–1318.

[S9] 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.

[S10] S. Güttel and M. Schweitzer, *Randomized sketching for Krylov approximations of large-scale matrix functions*, SIAM J. Matrix Anal. Appl., 44 (2023), pp. 1073–1095.

[S11] P. Kandolf, A. Koskela, S. D. Relton, and M. Schweitzer, *Computing low-rank approximations of the Fréchet derivative of a matrix function using Krylov subspace methods*, Numer. Linear Algebra Appl., 28 (2021), p. e2401.

[S12] K. Lund and M. Schweitzer, *The Fréchet derivative of the tensor t-function*, Calcolo, 60 (2023), p. 35.

[S13] M. Schweitzer, *Decay bounds for Bernstein functions of Hermitian matrices with applications to the fractional graph Laplacian*, Electron. Trans. Numer. Anal., 55 (2022), pp. 438–454.

[S14] M. Schweitzer, *Integral representations for higher-order Fréchet derivatives of matrix functions: Quadrature algorithms and new results on the level-2 condition number*, Linear Algebra Appl., 656 (2023), pp. 247–276.

[S15] M. Schweitzer, *Sensitivity of matrix function based network communicability measures: Computational methods and a priori bounds*, SIAM J. Matrix Anal. Appl., 44 (2023), pp. 1321–1348.

CONTENTS

Acknowledgments **I**

1 Introduction **1**

1.1 Notation 1

1.2 Functions of matrices 2

1.3 Completely monotonic functions and related classes 7

1.4 Krylov subspace methods 12

2 Restarted and sketched Krylov methods for matrix functions **17**

2.1 Restarted Krylov subspace methods 17

2.2 Krylov methods with randomized sketching 31

3 Fréchet derivatives and low-rank updates of matrix functions **37**

3.1 Fréchet derivatives of matrix functions 38

3.2 Updating matrix functions subject to low-rank modifications . . . 54

3.3 Further comments on the connection between the two topics of this chapter 61

4 Decay bounds and probing methods **63**

4.1 Decay bounds for matrix functions 64

4.2 Probing methods for trace estimation 76

5 Contributions to each publication by the author **81**

CHAPTER 1

INTRODUCTION

In this chapter, we review basic properties of functions of matrices and briefly discuss selected application areas in which their computation occurs. After that, we introduce several classes of functions that play an important role in many of the publications discussed throughout Chapters 2 to 4. In the last section of this chapter, we give a brief introduction to Krylov subspace methods, mainly to fix terminology and notation.

None of the sections of this chapter aim to give a comprehensive overview of the respective area. Instead, the focus is on introducing the main, recurring concepts and on fixing notations that are used throughout this thesis. Where appropriate, we give pointers to relevant literature for further reading.

1.1 Notation

Before starting with the actual content, we introduce some notation that we frequently use.

The fields of real and complex numbers are denoted by $\mathbb{R}$ and $\mathbb{C}$, respectively. We denote by $\mathbb{K}^n$ the space of length-n vectors over the field $\mathbb{K}$ and by $\mathbb{K}^{m \times n}$ the space of matrices with m rows and n columns with entries in $\mathbb{K}$.

The set of polynomials (with complex coefficients) of degree at most d is denoted by Π_d.

Matrices are denoted by upper-case letters, vectors are denoted by lower-case bold letters and scalars are denoted by regular lower-case (greek or roman) letters.

The identity matrix of size $n \times n$ is denoted by I_n, or simply by I if its size is clear from the context. The ith canonical unit vector in $\mathbb{R}^n$ (i.e., the ith column of I_n) is denoted by $\boldsymbol{e}_i$. The symbols $\mathbf{0}$ and $\boldsymbol{1}$ denote vectors with all entries equal to zero or equal to one, respectively.

We denote the Euclidean vector norm (and the matrix norm it induces) by $\|\cdot\|$, and the Frobenius matrix norm by $\|\cdot\|_F$.

For a matrix A, its (i,j)-entry is denoted by $[A]_{ij}$. The transpose of a matrix A or a vector $\boldsymbol{v}$ is denoted by A^T or $\boldsymbol{v}^T$ and its complex adjoint by A^H or $\boldsymbol{v}^H$.

The field of values (or numerical range) of a square matrix A is denoted by $W(A) := \{\boldsymbol{x}^H A \boldsymbol{x} : \|\boldsymbol{x}\| = 1\}$.

1.2 Functions of matrices

When using the term *matrix function* in this thesis, we mean by this the extension of a scalar function $f : \Omega \to \mathbb{C}$, $\Omega \subseteq \mathbb{C}$ to square matrix arguments $A \in \mathbb{C}^{n \times n}$, i.e., $f : \mathcal{M} \to \mathbb{C}^{n \times n}$, $\mathcal{M} \subseteq \mathbb{C}^{n \times n}$ and we will always use the same symbol f for the scalar function and the corresponding matrix function. Such a matrix function can be defined in a variety of different ways, the three most common ones being via the Jordan canonical form of A, via a relation to Hermite interpolation and via the Cauchy integral formula; see [94, Definitions 1.2, 1.4 and 1.11]. When applicable, all these definitions are mathematically equivalent [94, Theorem 1.12]. We recall the latter two of these three ways of defining a matrix function in detail below, as they form the basis of many results contained in this thesis.

Definition 1.1. Let $A \in \mathbb{C}^{n \times n}$ and denote by $\lambda_i, i = 1, \ldots, s$ the distinct eigenvalues of A and by $n_i, i = 1, \ldots, s$ the size of the largest Jordan block of A corresponding to λ_i. Then f is said to be *be defined on the spectrum of A* if the values

$$f^{(j)}(\lambda_i), \qquad j = 0, \ldots, n_i - 1, \quad i = 1, \ldots, s,$$

exist.

If f is defined on the spectrum of A, then the *matrix function* $f(A)$ is defined via $f(A) := p(A)$, where p interpolates f on the spectrum of A in the Hermite sense, i.e., it is the unique polynomial $p \in \Pi_{\deg \psi - 1}$, where ψ denotes the minimal polynomial of A, that fulfills

$$p^{(j)}(\lambda_i) = f^{(j)}(\lambda_i), \qquad j = 0, \ldots, n_i - 1, \quad i = 1, \ldots, s.$$

Definition 1.2. Let $A \in \mathbb{C}^{n\times n}$ and let f be analytic on and inside a closed contour Γ that winds around $\operatorname{spec}(A)$, the spectrum of A, exactly once in counterclockwise direction. Then the *matrix function* $f(A)$ is defined via

$$f(A) := \frac{1}{2\pi i}\int_\Gamma f(z)(zI - A)^{-1}\,\mathrm{d}z. \tag{1.1}$$

Matrix functions play an important role in many applications in science and engineering, of which some frequently occur as model problems in our publications that are summarized in this thesis. We therefore introduce them with a bit of detail in the following examples.

Example 1.3. Probably the most important and prominent matrix function (besides the matrix inverse $f(A) = A^{-1}$) is the matrix exponential, $\exp(A)$. One of its many applications is in the solution of ordinary differential equations (ODEs) via *exponential integration schemes*: By the variation-of-constants formula, the semilinear equation

$$\boldsymbol{u}'(t) + A\boldsymbol{u}(t) = g(\boldsymbol{u}(t)), \qquad \boldsymbol{u}(0) = \boldsymbol{u}_0 \tag{1.2}$$

is solved by

$$\boldsymbol{u}(t) = \exp(-tA)\boldsymbol{u}_0 + \int_0^t \exp(-(t-\tau)A)g(\boldsymbol{u}(\tau))\,\mathrm{d}\tau. \tag{1.3}$$

Numerical methods for solving (1.2) now arise by appropriately approximating the integral on the right-hand side of (1.3). E.g., using a simple rectangular rule with step size h, one obtains the approximation

$$\boldsymbol{u}_1 = \exp(-hA)\boldsymbol{u}_0 + h\varphi(-hA)g(\boldsymbol{u}_0),$$

for $\boldsymbol{u}_1 \approx \boldsymbol{u}(h)$, where

$$\varphi(z) = \frac{\exp(z) - 1}{z}. \tag{1.4}$$

This is the *exponential Euler scheme*. Other exponential integration schemes arise by using more sophisticated quadrature rules for the integral; see, e.g., [101, 102] for an overview.

Example 1.4. Another application of the matrix exponential arises in the analysis of complex networks represented by graphs. Let $G = (\mathcal{V}, \mathcal{E})$ be an undirected graph, where $\mathcal{V}$ denotes the set of nodes, $\mathcal{E}$ denotes the set of edges and $|\mathcal{V}| = n$. For notational simplicity, we number the nodes such that we can take $\mathcal{V} = \{1, \ldots, n\}$. The adjacency matrix A_G of G is a binary matrix with entries

$$[A_G]_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases}$$

It is a well-known fact that powers of the adjacency matrix encode information about walks in the graph G. To be specific, $[A_G^k]_{ij}$ equals the number of *length-k walks* in G that connect node i and node j. Thus, taking the power series definition of the matrix exponential,

$$\exp(A_G) = \sum_{k=0}^{\infty} \frac{1}{k!} A_G^k,$$

one can interpret the entry $[\exp(A_G)]_{ij}$ as a weighted sum of the number of all walks between two nodes i and j, where the influence of a walk is discounted according to the factorial of its length.

Due to the intuition that it is easier to spread information (or some other quantity) through a network if there exist many different short paths between nodes, the entries of $\exp(A_G)$ play an important role in characterizing the *communicability* of the network. In particular, in [75], it is proposed to use the diagonal entries $[\exp(A_G)]_{ii}$—which indicate how easy it is for information that was sent out from node i to return to its source—as a measure of the importance of nodes for the overall ability of the network to spread information. This measure is called the *(exponential) subgraph centrality* of node i.

Related is the *Estrada index* of the graph G,

$$\mathrm{EE}(G) := \mathrm{tr}(\exp(A_G)) = \sum_{i=1}^{n} [\exp(A_G)]_{ii},$$

the sum of all subgraph centralities. It measures the overall communicability of the network and also has applications, e.g., in protein folding [71, 73].

As computing the trace of an implicitly given matrix function can be a computationally intensive task, which, e.g., requires the use of stochastic estimators [103, 131], alternative measures have also been suggested. A prominent alternative is the *total communicability* of the network,

$$\mathrm{TC}(G) := \boldsymbol{1}^H \exp(A_G) \boldsymbol{1} = \sum_{i=1}^{n} \sum_{j=1}^{n} [\exp(A_G)]_{ij}, \tag{1.5}$$

introduced in [19]. In contrast to the Estrada index, walks between any pair of nodes—instead of only walks connecting a node to itself—contribute to this measure.

For further discussion of centrality and communicability measures, also based on other matrix functions than the exponential, we refer the reader to the survey [17] and the references therein.

Example 1.5. Fractional powers of matrices arise, e.g., in the context of modeling nonlocal dynamics on complex networks. Let A_G be the adjacency matrix of a complex network (again modeled as a graph G) and let $L_G = D_G - A_G$, with $[D_G]_{ii} = \deg(i)$, be the corresponding graph Laplacian, which is singular and positive semidefinite.

The *fractional graph Laplacian* L_G^α with $\alpha \in (0,1)$ has recently emerged as a tool for describing nonlocal phenomena like anomalous diffusion and so-called Lévy flights [16, 25, 32, 33, 72].

One can show that $\lambda = 0$ is a semisimple eigenvalue of L_G, so that its fractional powers are well-defined [16, Proposition 2.4 & Theorem 2.5]. As L_G is a singular M-matrix, L_G^α is also a singular M-Matrix, which in particular implies that its diagonal is positive and all its offdiagonal entries are nonpositive. Therefore, L_G^α can be interpreted as the Laplacian of a weighted, fully connected network with the same node set as G, i.e., it introduces "long-distance connections" between nodes that were originally not connected in G. The magnitude of the entries of L_G^α thus provides information about the strength or likelihood of these nonlocal effects and is therefore of interest for understanding these phenomena.

In our publication [S13], which is discussed in Section 4.1.3, we derive bounds for the size of the entries of the fractional Laplacian.

Example 1.6. In lattice quantum chromodynamics (QCD), the theory describing the strong interaction between quarks and gluons is simulated on a four-dimensional space-time lattice. The most important relation for describing this interaction is the Dirac equation [59]

$$(\mathcal{D} + m)\psi(x) = \eta(x), \tag{1.6}$$

where m is a scalar parameter, ψ and η represent quark fields (where $\psi(x), \eta(x)$ are vectors with twelve entries, corresponding to all combinations of three colors and four spins) and $\mathcal{D}$ is the *Dirac operator*

$$\mathcal{D} = \sum_{i=0}^{3} \gamma_i \otimes \left(\frac{\partial}{\partial x_i} + A_i \right)$$

with $A_i(x) \in \mathbb{C}^{3\times 3}$ elements of the Lie algebra $\mathfrak{su}(3)$ of the special unitary group SU(3) and $\gamma_i \in \mathbb{C}^{4\times 4}, i = 0, \ldots, 3$ generators of the Clifford algebra $C\ell_4(\mathbb{C})$.

For simulations, the Dirac equation (1.6) is discretized on an $N_t \times N_s^3$ lattice with uniform lattice spacing a and N_t and N_s denoting the number of lattice points in the time dimension and in each of the three spatial dimensions, respectively. Using the Wilson discretization [168] with periodic boundary conditions results in

$$\begin{aligned}(D_W\phi)(x) &= \frac{m_0+4}{a}\phi(x) - \frac{1}{2a}\sum_{i=0}^{3}\big((I_4-\gamma_i)\otimes U_i(x)\big)\phi(x+a\boldsymbol{e}_i) \\ &\quad -\frac{1}{2a}\sum_{i=0}^{3}\big((I_4+\gamma_i)\otimes U_i^H(x-a\boldsymbol{e}_i)\big)\phi(x-a\boldsymbol{e}_i), \qquad (1.7)\end{aligned}$$

where the parameter m_0 sets the quark mass, and the gauge links $U_i(x) \in \mathbb{C}^{3\times 3}$ are elements of SU(3).

The Wilson–Dirac operator (1.7) is Γ_5-symmetric, i.e.,

$$(\Gamma_5 D_W)^H = \Gamma_5 D_W \quad \text{with} \quad \Gamma_5 = I_{N_t N_s^3} \otimes \gamma_0\gamma_1\gamma_2\gamma_3 \otimes I_3;$$

see, e.g., [82]. For certain simulations, it is important that the discretized operator fulfills the Ginsparg–Wilson relation

$$\Gamma_5 D + D\Gamma_5 = aD\Gamma_5 D,$$

a lattice version of chiral symmetry [87]. This is not fulfilled by the Wilson–Dirac operator, but by the related *Neuberger overlap operator* [134],

$$D_N = \rho I + \Gamma_5 \operatorname{sign}(\Gamma_5 D_W), \text{ where } \rho > 1, \qquad (1.8)$$

which involves the matrix sign function. It is common practice to use the identity $\operatorname{sign}(A) = A(A^2)^{-1/2}$, to rewrite the sign function in terms of the Cauchy–Stieltjes function (cf. Section 1.3) $f(z) = z^{-1/2}$. In simulations, one has to solve linear systems with the overlap operator, for which one typically uses a Krylov subspace method (cf. Section 1.4). In each iteration of such a method, one needs to approximate the action of the sign function (or the inverse square root) on a vector.

In the presence of a nonzero *chemical potential* ν, the links in time direction in the Wilson–Dirac operator change, resulting in a discretized operator D_W^ν which is *not* Γ_5-symmetric, such that the solution of systems with the overlap operator (1.8) now involves approximating the matrix sign function (or inverse square root) of a non-Hermitian matrix.

In all of the above applications that involve a matrix function $f(A)$, the matrix A to which f is applied is *large and sparse*. It is important to note that $f(A)$ does not inherit this structural property and is in general a full matrix, even when A is extremely sparse.[1] This has many important consequences, e.g., that for large n the storage that would be required for $f(A)$ is immense, so that explicitly forming it is impossible even if one could afford the large computational cost. Luckily, many applications—including those in Examples 1.3 to 1.6 above—do not actually require the matrix function $f(A)$ itself, but rather its action on a vector $f(A)\boldsymbol{b}$, a quadratic form $\boldsymbol{b}^H f(A)\boldsymbol{b}$ or some other quantity like its trace that does not require storing the whole matrix. These quantities can often be approximated efficiently by iterative methods at cost and storage much smaller than what would be necessary for $f(A)$; cf. Section 1.4 for a brief discussion of Krylov subspace methods that are often used for this purpose.

1.3 Completely monotonic functions and related classes

From the examples we hinted at in Section 1.2 it already becomes apparent that a wide variety of different functions of matrices are of practical interest. It is therefore quite obvious that many important statements and properties cannot be expected to be valid for all matrix functions, but only for individual functions or classes of functions sharing certain characteristics.

A common way of obtaining results that apply to a quite broad range of functions is by considering function classes that allow to transfer or extend results from *the* two prototypical matrix functions, the inverse and the exponential. As these two functions are arguably by far the most important and common, they are of great interest in their own right and very well studied, so that there exists a wide variety of specialized algorithmic and theoretical techniques for them.

[1] Except for a few special cases: E.g., when A is (block) diagonal or (block) triangular, so is $f(A)$.

A straight-forward approach for linking general analytic matrix functions to the inverse function arises from the definition in terms of the Cauchy integral formula (1.1). This connection can be exploited both algorithmically [81] and to obtain theoretical results [47]. This approach has some shortcomings which sometimes limit its applicability, though. E.g., even when A is Hermitian positive definite, the matrices $zI - A$ for $z \in \Gamma$ do not inherit this property, as $\Gamma \not\subset \mathbb{R}$ in general. Thus, taking this route, theoretical results for the inverse of Hermitian positive definite matrices cannot be transferred to f evaluated at such matrices.

Due to this fact, it is common to restrict to more specific subsets of analytic functions. Interestingly, the function classes that naturally arise in this setting have also been studied intensively in several branches of pure mathematics for more than a century. All of them are in some way related to the concept of *complete monotonicity* [4, 27, 28, 147].

Definition 1.7. A function $f : \mathbb{R}^+ \longrightarrow \mathbb{R}$ is called *completely monotonic* if it is infinitely many times continuously differentiable and satisfies

$$(-1)^k f^{(k)}(z) \geq 0 \text{ for } k \in \mathbb{N}_0 \text{ and } z \in \mathbb{R}^+.$$

The first class of functions we consider is the class of *(Cauchy–)Stieltjes functions*, sometimes also called *Markov functions*.

Definition 1.8. Let μ be a monotonically increasing, real-valued function on $\mathbb{R}_0^+$ such that

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

and let $a \geq 0$. Then the function $f : \mathbb{C} \setminus \mathbb{R}_0^- \longrightarrow \mathbb{C}$ defined via

$$f(z) = a + \int_0^\infty \frac{1}{z+t}\,\mathrm{d}\mu(t) \tag{1.9}$$

is called *(Cauchy–)Stieltjes function corresponding to μ.*

It is well known that the derivatives of a Cauchy–Stieltjes function are given by

$$f^{(k)}(z) = (-1)^k k! \int_0^\infty \frac{1}{(z+t)^{k+1}}\,\mathrm{d}\mu(t) \text{ for all } k \in \mathbb{N}_0,$$

see, e.g., [27, Section 3]. From this, it is apparent that every Cauchy–Stieltjes function[2] is completely monotonic.

[2] Or more precisely, its restriction to $\mathbb{R}^+$.

Cauchy–Stieltjes function arise in a variety of applications, and several well-known functions have a representation of the form (1.9). A few examples of important functions belonging to this class are ...

- ...the inverse $f(z) = z^{-1}$, generated by the step function
$$\mu(t) = \begin{cases} 0 & t = 0, \\ 1 & t > 0, \end{cases}$$
- ...inverse fractional powers $f(z) = z^{-\alpha}$ for $\alpha \in (0, 1)$, because
$$z^{-\alpha} = \frac{\sin(\alpha\pi)}{\pi} \int_0^\infty \frac{t^{-\alpha}}{z+t}\,\mathrm{d}t,$$
- ...the function $f(z) = \log(1+z)/z$, because
$$\frac{\log(1+z)}{z} = \int_1^\infty \frac{t^{-1}}{z+t}\,\mathrm{d}t.$$

When f is a Cauchy–Stieltjes function, then $f(A)$ can naturally be represented as
$$f(A) = \int_0^\infty (A + tI)^{-1}\,\mathrm{d}\mu(t). \tag{1.10}$$
Formally, this can be shown by starting from the Cauchy integral formula (1.1) with a suitable contour Γ and inserting the Stieltjes representation,
$$\begin{aligned} f(A) &= \frac{1}{2\pi i}\int_\Gamma f(z)(zI - A)^{-1}\,\mathrm{d}z \\ &= \frac{1}{2\pi i}\int_\Gamma \left(\int_0^\infty \frac{1}{z+t}\,\mathrm{d}\mu(t)\right)(zI - A)^{-1}\,\mathrm{d}z \\ &= \int_0^\infty \left(\frac{1}{2\pi i}\int_\Gamma \frac{1}{z+t}(zI - A)^{-1}\,\mathrm{d}z\right)\mathrm{d}\mu(t) \\ &= \int_0^\infty (A + tI)^{-1}\,\mathrm{d}\mu(t), \end{aligned}$$
where the last equality follows from the Cauchy integral representation of the resolvent.

When A is Hermitian positive definite, then so are all the shifted matrices $A + tI$ appearing in the Stieltjes representation of (1.10). This fact has been exploited for obtaining a wide variety of theoretical results for Stieltjes matrix functions, see, e.g., [23, 24, 80, 91, 126] and the papers [S6, S9, S10] discussed in this thesis.

Another closely related class of functions is the set of *Laplace–Stieltjes functions.*

Definition 1.9. Let σ be a monotonically increasing, real-valued function on $\mathbb{R}_0^+$ and let $a \geq 0$. Then the function f defined via

$$f(z) = a + \int_0^\infty \exp(-\tau z) \, \mathrm{d}\sigma(\tau).$$

is called *Laplace–Stieltjes function corresponding to* σ or the *Laplace transform* of the measure $\mathrm{d}\sigma$.

In fact, one can show that every completely monotonic function is a Laplace–Stieltjes function.[3] This is known as *Bernstein's theorem*; see, e.g., [147, Theorem 1.4].

A particular implication of Bernstein's theorem is that every Cauchy–Stieltjes function is also a Laplace–Stieltjes function. One can make this connection even more explicit: Using the well-known identity

$$\int_0^\infty \exp(-\tau z) \exp(-\tau t) \, \mathrm{d}\tau = \frac{1}{t+z}$$

for the Laplace transform of the exponential, together with Fubini's theorem, one can write a Stieltjes function as

$$\begin{aligned} f(z) &= \int_0^\infty \frac{1}{t+z} \, \mathrm{d}\mu(t) \\ &= \int_0^\infty \left(\int_0^\infty \exp(-\tau z) \exp(-\tau t) \, \mathrm{d}\tau \right) \mathrm{d}\mu(t) \\ &= \int_0^\infty \exp(-\tau z) \left(\int_0^\infty \exp(-\tau t) \, \mathrm{d}\mu(t) \right) \mathrm{d}\tau \\ &= \int_0^\infty \exp(-\tau z) \, \mathrm{d}\sigma(\tau), \end{aligned}$$

where

$$\mathrm{d}\sigma(\tau) = g(\tau) \, \mathrm{d}\tau \quad \text{with} \quad g(\tau) = \int_0^\infty \exp(-\tau t) \, \mathrm{d}\mu(t).$$

Thus, Cauchy–Stieltjes functions are *"double"* Laplace–Stieltjes functions. Just as Cauchy–Stieltjes functions allow to transfer results for the inverse to more general $f(A)\boldsymbol{b}$, Laplace–Stieltjes functions allow the same for results concerning the exponential. This has, e.g., been exploited in [23,24,126] and our own work [S5].

The last class of functions that plays an important role in this thesis is the class of *Bernstein functions.* While not being completely monotonic themselves, they

[3] The classes actually coincide when one poses certain natural conditions on the admissible measures.

can be loosely characterized as “positive primitives of completely monotonic functions”.

Definition 1.10. A function $f : \mathbb{R}^+ \to \mathbb{R}$ is called *Bernstein function* if it is infinitely many times continuously differentiable, nonnegative and satisfies

$$(-1)^{k-1} f^{(k)}(z) \geq 0 \text{ for } k \in \mathbb{N} \text{ and } z \in \mathbb{R}^+.$$

Each Bernstein function exhibits the so-called *Lévy–Khintchine representation*

$$f(z) = a + bz + \int_0^\infty (1 - \exp(-tz)) \, \mathrm{d}\lambda(t) \tag{1.11}$$

where $a, b \geq 0$ and λ is a positive measure (the *Lévy measure*) on $(0, \infty)$ such that

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

see, e.g., [27, 147]. Additionally, any Bernstein function admits a continuous extension to the origin.

Important examples of Bernstein functions are ...

- ...the function $f(z) = 1 - e^{-tz}$, $t \geq 0$, corresponding to the Lévy measure

$$\lambda(t) = \begin{cases} 0 & t = 0, \\ 1 & t > 0, \end{cases}$$

- ...fractional powers $f(z) = z^\alpha$, $\alpha \in (0, 1)$, because

$$z^\alpha = \frac{\alpha}{\Gamma(1-\alpha)} \int_0^\infty (1 - \exp(-tz)) t^{-(\alpha+1)} \, \mathrm{d}t$$

- ...and the (shifted) logarithm $f(z) = \log(1 + z)$.[4]

Just as for Laplace–Stieltjes functions, one can use the Lévy–Khintchine representation to transfer results from the exponential function to Bernstein functions. Examples of this technique can be found in our papers [S5, S13].

[4] Note that no closed-form expression of the Lévy measure of this function is known.

1.4 Krylov subspace methods

Krylov subspace methods are certainly the most frequently used class of methods for approximating $f(A)\boldsymbol{b}$ the action of a matrix function on a vector, or a quadratic form $\boldsymbol{b}^H f(A)\boldsymbol{b}$. For linear systems and eigenvalue problems, these methods haven been studied since the 1950s [6,93,120], while the early references for more general f date back to the late 1980s and early 1990s [62,113,144].

In this section, we give a brief overview of the main ideas behind these methods, in order to introduce the most important concepts and fix our notation. The treatment is necessarily far from being exhaustive and we refer to, e.g., [94, Chapter 13] or [89] and the references therein for more details on both polynomial and rational Krylov methods.

We begin by defining what a Krylov subspace is.

Definition 1.11. Let $A \in \mathbb{C}^{n\times n}$ and let $\boldsymbol{b} \in \mathbb{C}^n$. The *mth (polynomial) Krylov subspace* with respect to A and $\boldsymbol{b}$ is defined as

$$\mathcal{K}_m(A, \boldsymbol{b}) = \{p_{m-1}(A)\boldsymbol{b} : p_{m-1} \in \Pi_{m-1}\} = \operatorname{span}\{\boldsymbol{b}, A\boldsymbol{b}, \ldots, A^{m-1}\boldsymbol{b}\}.$$

The standard way of extracting an approximation for $f(A)\boldsymbol{b}$ from a Krylov subspace is by using a Rayleigh–Ritz procedure. Given a matrix $W_m \in \mathbb{C}^{n\times m}$ whose columns $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_m$ form a basis of the Krylov subspace, the Rayleigh-Ritz approximation is defined as

$$\boldsymbol{f}_m := W_m f(W_m^\dagger A W_m) W_m^\dagger \boldsymbol{b}, \tag{1.12}$$

where $W_m^\dagger$ denotes the *Moore-Penrose pseudoinverse* [13] of W_m. Typically, due to numerical stability considerations, one works with an *orthonormal basis (ONB)* $V_m = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m]$, which in the polynomial case is computed by Arnoldi's method [6]. Collecting the coefficients of the modified Gram–Schmidt orthonormalization in an upper Hessenberg matrix H_m, the involved matrices fulfill the *Arnoldi relation*

$$AV_m = V_m H_m + h_{m+1,m}\boldsymbol{v}_{m+1}\boldsymbol{e}_m^H. \tag{1.13}$$

An important consequence of (1.13) is that $V_m^H A V_m = H_m$. As $V_m^H \boldsymbol{b} = \|\boldsymbol{b}\|\boldsymbol{e}_1$ and $V_m^\dagger = V_m^H$ because V_m has orthonormal columns, the Rayleigh-Ritz approximation simplifies to

$$\boldsymbol{f}_m = \|\boldsymbol{b}\| V_m f(H_m)\boldsymbol{e}_1 \tag{1.14}$$

and is called *mth Arnoldi approximation for $f(A)\boldsymbol{b}$*. When A is Hermitian, the Arnoldi basis vectors fulfill a three-term recurrence, so that (at least in exact arithmetic) it is only necessary to orthogonalize each new basis vector against the two previous vectors and, accordingly, H_m is tridiagonal.[5] The resulting method is the so-called *Lanczos process* [120].

By [144, Theorem 3.3] it is known that $\boldsymbol{f}_m = p_{m-1}^*(A)\boldsymbol{b}$, where $p_{m-1}^* \in \Pi_{m-1}$ is the unique polynomial interpolating f at the eigenvalues of H_m (the *Ritz values*) in the Hermite sense. It directly follows that $\boldsymbol{f}_m = f(A)\boldsymbol{b}$ whenever f is a polynomial of degree at most $m-1$.

This also has important consequences for obtaining convergence results for Krylov methods[6] from polynomial approximation results. For *any* $p \in \Pi_{m-1}$, we have

$$\begin{aligned}\|f(A)\boldsymbol{b} - \boldsymbol{f}_m\| &= \big\|f(A)\boldsymbol{b} - p(A)\boldsymbol{b} + \|\boldsymbol{b}\|V_m p(H_m)\boldsymbol{e}_1 - \|\boldsymbol{b}\|V_m f(H_m)\boldsymbol{e}_1\big\| \\ &\leq \|\boldsymbol{b}\| \cdot \big(\|f(A) - p(A)\| + \|p(H_m)\boldsymbol{e}_1 - f(H_m)\boldsymbol{e}_1\|\big) \\ &\leq 2C\|\boldsymbol{b}\| \cdot \left(\min_{p^* \in \Pi_{m-1}} \max_{z \in W(A)} |f(z) - p^*(z)|\right), \qquad (1.15)\end{aligned}$$

with $C = 1$ if A is normal and $C = 1+\sqrt{2}$ otherwise. The last inequality (1.15) is a result of the Crouzeix-Palencia theorem [49,51] and the fact that $W(H_m) \subseteq W(A)$. Inequality (1.15) is also known as the *quasi-optimality property* of the Arnoldi approximation.

From a computational perspective, the main work required for the Arnoldi approximation (1.14) consists of m matrix-vector products with A (which typically have an overall cost of $\mathcal{O}(mn)$ when A is sparse with $\mathcal{O}(n)$ nonzeros) and the modified Gram–Schmidt orthonormalization, which requires $2m$ inner products (i.e., computational cost $\mathcal{O}(mn)$) for Hermitian A and $\frac{m(m-1)}{2}$ inner products (i.e., computational cost $\mathcal{O}(m^2n)$) for non-Hermitian A.

In cases where a large number m of Krylov iterations is necessary, e.g., when f is difficult to approximate by a low-degree polynomial on $W(A)$, the computational cost for the orthogonalization quickly becomes very high for nonsymmetric A (even more so when working in a highly parallel computing environment, where each inner product requires global communication), which can severely limit the practical applicability of the method. Additionally, evaluating (1.14) requires the storage of the full Krylov basis V_m,[7] which can quickly exceed the available capacity for large scale problems. E.g., when $n = 10^7$ and IEEE double precision

[5] For the sake of a unified presentation, we refrain from changing the notation from H_m to T_m in the Hermitian case, as it is frequently done in the literature.

[6] Strictly speaking, “convergence” is not the correct term here, as a Krylov method is guaranteed to terminate with the exact vector $f(A)\boldsymbol{b}$ after a finite number of steps (at least in the unrestarted case). We still use this term in accordance with the literature.

[7] This is in contrast to the linear system case, where the short recurrence for the basis vectors turns into a short recurrence for the Arnoldi approximations, yielding the well-known conjugate gradient algorithm [93] when A is positive definite, or the SYMMLQ method [135] when A is indefinite.

is used, each basis vector requires 80 MB of storage, so that on a system with 32 GB RAM, no more than 400 vectors can be kept in memory at the same time.

There are two possible remedies for these computational limitations. Either, one uses other approximation spaces with more powerful approximation properties, as, e.g., *rational Krylov subspaces* [63, 69, 89–91, 107, 108, 126, 132, 142, 143, 149] which we briefly discuss below, or one resorts to *restarted* Krylov methods [1, 40, 41, 65, 66, 80, 81, 105, 160] which perform several *cycles* in which a Krylov space of (small) dimension at most $m_{\max}$ is built. These methods typically delay convergence compared to a "full" Krylov method but require a lot less computational work per iteration. Restarted methods are described in more detail in Section 2.1 where we discuss some of our own recent contributions to the area.

Rational Krylov methods work similarly to standard (polynomial) ones, simply using a different approximation space.

Definition 1.12. Let $A \in \mathbb{C}^{n\times n}$, $\boldsymbol{b} \in \mathbb{C}^n$ and fix a polynomial $q_{m-1} \in \Pi_{m-1}$. The *mth rational Krylov subspace with respect to A, $\boldsymbol{b}$ and q_{m-1}* is defined as

$$\begin{aligned} \mathcal{Q}_m(A,\boldsymbol{b}) &= \{r_{m-1}(A)\boldsymbol{b} : r_m = p_{m-1}/q_{m-1}, p_{m-1} \in \Pi_{m-1}\} \\ &= q_{m-1}(A)^{-1}\mathcal{K}_m(A,\boldsymbol{b}). \end{aligned} \tag{1.16}$$

The zeros $\xi_1, \ldots, \xi_{m-1}$ of the polynomial q_{m-1} in (1.16) are called the *poles* of the rational Krylov space.[8]

An approximation from a rational Krylov space can be computed analogously to (1.12), just that the columns of W_m now form a basis of $\mathcal{Q}_m(A,\boldsymbol{b})$ instead of $\mathcal{K}_m(A,\boldsymbol{b})$. Similarly to the polynomial case, an orthonormal basis can be computed by the *rational Arnoldi method* [142]. The corresponding rational Arnoldi relation is a bit more involved than its polynomial counterpart (1.13),

$$A(V_mK_m + h_{m+1,\xi_m^{-1}}\boldsymbol{v}_{m+1}\boldsymbol{e}_m^H) = V_mH_m + h_{m+1,m}\boldsymbol{v}_{m+1}\boldsymbol{e}_m^H,$$

where H_m is again upper Hessenberg and $K_m = I_m + H_m \operatorname{diag}(\xi_1^{-1}, \ldots, \xi_m^{-1})$.[9] In particular, the compressed matrix $V_m^HAV_m$ is *not* given by the matrix containing the orthogonalization coefficients any longer. If, however, $\xi_m = \infty$, one has $V_m^HAV_m = H_mK_m^{-1}$ so that it can be easily computed working only with matrices of size m; see, e.g., [89, Lemma 5.6].

[8] Note that we allow that some of the poles ξ_i may be infinite.

[9] There exist variants of the rational Arnoldi method using so-called *continuation vectors* which yield a different K_m in which I_m is replaced by an upper triangular matrix, but we ignore these variants here.

The main computational work in an iteration of a rational Krylov subspace method lies in the solution of a shifted linear system of the form $(A+\xi_i I)\boldsymbol{x} = \boldsymbol{y}$, which typically is much more costly than the matrix vector product and orthogonalization that are also required. Rational Krylov methods are particularly well-suited when (shifted) linear systems with A can be efficiently solved by a sparse direct solver. In that case, it is common practice to cyclically repeat the poles so that factorizations can be reused. When it is not possible to solve linear systems by a direct solver, one has to resort to so-called *inner-outer* Krylov schemes, where the linear systems for the (outer) rational method are solved by another Krylov method (the inner method). We discuss such approaches in Section 2.1.2 in the context of our publication [S9].

Two important special cases of rational Krylov methods arise from restricting to one or two repeated poles: Alternatingly choosing poles $\xi_{2i-1} = 0, \xi_{2i} = \infty, i = 1, 2, \ldots$ [10] yields so-called *extended Krylov subspace methods* [63, 107, 108, 149, 153, 154], while choosing a single repeated pole $\xi_i \equiv \xi$ for all i yields the *shift-invert* or *RD-rational*[11] Krylov methods [69, 132]. As it is typically implemented in a different way than “standard” rational Krylov methods, we briefly recall the usual way of defining the shift-invert Krylov approximation. Defining $B := (A-\xi I)^{-1}$, one builds an ONB V_m of the Krylov subspace $\mathcal{K}_m(B, \boldsymbol{b})$ by the usual Arnoldi method. Then, an Arnoldi approximation

$$\boldsymbol{g}_m := \|\boldsymbol{b}\| V_m g(H_m)\boldsymbol{e}_1$$

with $g(y) = f(y^{-1}+\xi)$ is extracted. Note that g is chosen such that $g(B)\boldsymbol{b} = f(A)\boldsymbol{b}$.

Note that for general pole sequences, even for Hermitian A, the basis vectors do not fulfill a short-term recurrence. In [136], a short recurrence rational Krylov method is introduced, which, however, requires the solution of a linear system with *two* right-hand sides in each iteration. Clearly the shift-invert method inherits the three-term recurrence from polynomial Krylov spaces when A is Hermitian, as it uses the standard Arnoldi/Lanczos method, while for extended Krylov spaces, the basis vectors fulfill a five-term recurrence in the Hermitian case [107, 108].

Note that for convergence analysis of rational Krylov methods, one can repeat the derivation presented in (1.15) for polynomial Krylov spaces, but replace the polynomial p^* by a rational function with denominator polynomial q_{m-1} (and arbitrary numerator polynomial). Compared to polynomials, rational functions typically require a much lower degree for similar approximation accuracy, so that rational Krylov methods tend to converge much more rapidly.

[10] Depending on the specific implementation, the order in which the poles 0 and ∞ occur might be different, and sometimes one might also, e.g., want to have more poles at ∞ than at zero.

[11] RD means “restricted denominator”.

Clearly, as the denominator polynomial q_{m-1} is fixed (while the numerator polynomial is chosen suitably by the method itself), the approximation quality (and thus speed of convergence) largely depends on a proper choice of poles, so that pole selection in rational Krylov methods has become a very active area of research. While beyond the scope of this thesis, we briefly mention a few important references: The two most commonly employed approaches are either based on black-box adaptive pole selection strategies [91] which automatically determine poles while running the method and strategies based on equidistributed sequences which yield quasi-optimal poles [64, 126].

CHAPTER 2

RESTARTED AND SKETCHED KRYLOV METHODS FOR MATRIX FUNCTIONS

As discussed in Section 1.4, one of the main limitations when using Krylov subspace methods for approximating the action of a very large matrix function $f(A)$ on a vector is the excessive growth of memory requirements and (when A is non-Hermitian) orthogonalization cost. To overcome this, restarts or rational Krylov methods can be used.

Throughout this chapter, we assume that we are in a setting in which it is *not possible* to efficiently solve (shifted) linear systems with A by a direct solver, e.g., due to size and structural properties, or even because A is not available as a matrix, but only implicitly through a routine that returns the result of a matrix vector product $\boldsymbol{x} \mapsto A\boldsymbol{x}$. In these cases, one is restricted to using restarted polynomial Krylov methods (or rational Krylov methods with inner polynomial solves). We discuss several of our contributions to these areas in Section 2.1 below.

As a very recent alternative, techniques from *randomized numerical linear algebra*, namely sketching and oblivious subspace embeddings [125,146,157,170], have been investigated as contenders for restarted methods. We cover these techniques in Section 2.2.

2.1 Restarted Krylov subspace methods

As briefly hinted at in Section 1.4, the goal of restarted Krylov methods is to approximate $f(A)\boldsymbol{b}$ to high accuracy while never building or storing a basis of

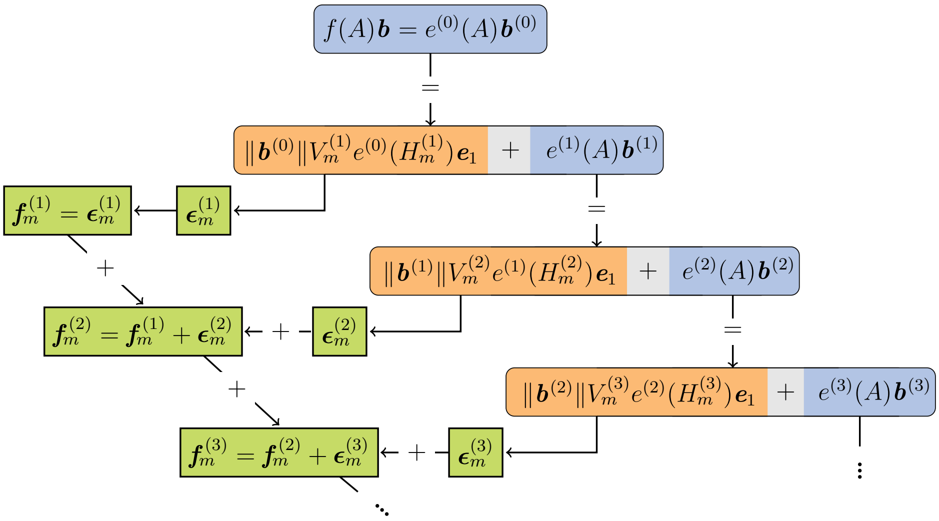

$f(A)\boldsymbol{b} = e^{(0)}(A)\boldsymbol{b}^{(0)}$
=
$\|\boldsymbol{b}^{(0)}\|V_m^{(1)}e^{(0)}(H_m^{(1)})\boldsymbol{e}_1$
+
$e^{(1)}(A)\boldsymbol{b}^{(1)}$
$\boldsymbol{f}_m^{(1)} = \boldsymbol{\epsilon}_m^{(1)}$
$\boldsymbol{\epsilon}_m^{(1)}$
=
$\|\boldsymbol{b}^{(1)}\|V_m^{(2)}e^{(1)}(H_m^{(2)})\boldsymbol{e}_1$
+
$e^{(2)}(A)\boldsymbol{b}^{(2)}$
+
$\boldsymbol{f}_m^{(2)} = \boldsymbol{f}_m^{(1)} + \boldsymbol{\epsilon}_m^{(2)}$
+
$\boldsymbol{\epsilon}_m^{(2)}$
=
+
$\|\boldsymbol{b}^{(2)}\|V_m^{(3)}e^{(2)}(H_m^{(3)})\boldsymbol{e}_1$
+
$e^{(3)}(A)\boldsymbol{b}^{(3)}$
$\boldsymbol{f}_m^{(3)} = \boldsymbol{f}_m^{(2)} + \boldsymbol{\epsilon}_m^{(3)}$
+
$\boldsymbol{\epsilon}_m^{(3)}$
$\vdots$
$\ddots$

Figure 2.1: Illustration of error function restarting.

a subspace of dimension larger than some small, prescribed dimension $m_{\max}$. In order to not restrict the accuracy of the method, this of course means that one typically needs to sequentially build several subspaces of dimension $m_{\max}$ in so-called *cycles* of the method. Restarting for matrix functions comes in two distinct flavors, which we will refer to as *error function restarting* and *residual-time restarting* in the following.

Error function restarting is reminiscent of how restarting for linear systems works (used, e.g., most prominently in the context of the GMRES method [145]). We briefly explain it for FOM (full orthogonalization method) in the following, which is mathematically equivalent to computing the Arnoldi approximation (1.14) for $f(z) = z^{-1}$.

Consider the linear system $A\boldsymbol{x} = \boldsymbol{b}$, assume that we have performed m steps of FOM, yielding an approximation $\boldsymbol{x}_m$ and define the residual $\boldsymbol{r}_m := \boldsymbol{b} - A\boldsymbol{x}_m$. By a direct computation, it fulfills the *residual equation*

$$A\boldsymbol{\epsilon}_m = \boldsymbol{r}_m, \tag{2.1}$$

where $\boldsymbol{\epsilon}_m := \boldsymbol{x} - \boldsymbol{x}_m$ denotes the *error* of the current approximation $\boldsymbol{x}_m$. Now, instead of progressing with the previous FOM iteration, one can instead discard all basis vectors computed so far and start a *second cycle* consisting of another m iterations of FOM for solving (2.1). The approximation $\boldsymbol{d}_m$ obtained from this cycle can then be added to $\boldsymbol{x}_m$ to obtain a new (and hopefully better) approximation for $\boldsymbol{x}$.

Restarting for more general matrix functions works very similarly, but with a few additional caveats. In case of FOM (or any other method for solving linear systems), the problem to solve in the second (and any subsequent) cycle is again a linear system with A, just with a different right-hand side. This does not stay valid for more general f. Instead, in the kth cycle of the method, one has to approximate

$$e^{(k)}(A)\boldsymbol{b}^{(k)},$$

the action of a matrix function $e^{(k)}(A)$ which is *different from* f on a vector $\boldsymbol{b}^{(k)}$ which is different from $\boldsymbol{b}$. Figure 2.1 illustrates the general principle behind error function restarting for $f(A)\boldsymbol{b}$.

Early work on restarting for matrix functions [65, 105, 160] represented the *error function* $e^{(k)}$ using divided differences, which unfortunately turns out to often be unstable in finite precision arithmetic. More stable restarts can be achieved by exploiting integral representations of f. In [81] we developed a restarted Krylov method for Cauchy–Stieltjes matrix functions (and related functions) based on this idea, which approximates the error function by adaptive numerical quadrature, as no numerically stable closed-form representation is known. We analyzed its convergence in [80]. Related to error function restarting, we discuss below a comparative study that we performed in [S9] in order to put this method into the context of other, competing approaches, as well as our recent extension of the ideas from [81] to Laplace transforms in [S5].

While error function restarting is quite broadly applicable to a large class of functions (though its practical usefulness depends on a stable error function representation which may not be available for all f), residual-time (RT) restarting is restricted to matrix function problems with an underlying differential equation. We briefly explain it here in the simplest possible setting, following [40]: Recall from Example 1.3 that $\exp(-A)\boldsymbol{b}$ is the solution of the initial value problem (IVP)

$$\boldsymbol{u}'(t) = -A\boldsymbol{u}(t), \qquad \boldsymbol{u}(0) = \boldsymbol{b} \tag{2.2}$$

at time $t = 1$. Residual-time restarting now approximates $\exp(-A)\boldsymbol{b}$ by combining a Krylov subspace method with a time-stepping scheme for (2.2): After performing m Arnoldi steps, an approximation

$$\boldsymbol{u}_m = \|\boldsymbol{b}\| V_m \exp(-t_0 H_m)\boldsymbol{e}_1 \approx \exp(-t_0 A)\boldsymbol{b}$$

for some suitably chosen time step $0 < t_0 \leq 1$ is computed (we describe below how to choose t_0). After doing so, one can discard V_m and approximate $\boldsymbol{u}(1) = \exp(A)\boldsymbol{b}$ as the solution of the modified IVP

$$\widetilde{\boldsymbol{u}}'(t) = -A\widetilde{\boldsymbol{u}}(t), \quad \widetilde{\boldsymbol{u}}(0) = \boldsymbol{u}_m \tag{2.3}$$

at time $1 - t_0$, i.e., $\widetilde{\boldsymbol{u}}(1 - t_0) \approx \exp(-A)\boldsymbol{b}$[12] by another m Krylov iterations. This

[12] Note that this is no equality, as the initial condition $\widetilde{\boldsymbol{u}}(0) = \boldsymbol{u}_m$ builds on the *inexact* Krylov approximation from the previous cycle.

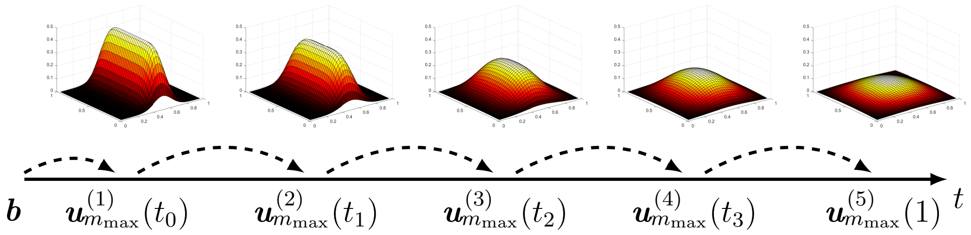

$\boldsymbol{b}$
$\boldsymbol{u}_{m_{\max}}^{(1)}(t_0)$
$\boldsymbol{u}_{m_{\max}}^{(2)}(t_1)$
$\boldsymbol{u}_{m_{\max}}^{(3)}(t_2)$
$\boldsymbol{u}_{m_{\max}}^{(4)}(t_3)$
$\boldsymbol{u}_{m_{\max}}^{(5)}(1)$
t

Figure 2.2: Schematic illustration of residual-time restarting.

process can be repeated until the final time $t = 1$ (for the original problem) is reached. Figure 2.2 contains a schematic illustration of this approach.

A crucial aspect for the success of this approach is the appropriate choice of the time step after each cycle. It can be based on the *residual* of $\boldsymbol{u}_m(t)$ from (2.3) with respect to (2.2), i.e.,

$$\boldsymbol{r}_m(t) := -\boldsymbol{u}_m'(t) - A\boldsymbol{u}_m(t). \tag{2.4}$$

It is well-known and straight-forward to verify that the residual (2.4) of a Krylov approximation $\boldsymbol{u}_m$ takes the form

$$\boldsymbol{r}_m(t) = \beta_m(t)\boldsymbol{v}_{m+1} \quad \text{where} \quad \beta_m(t) = -\|\boldsymbol{b}\| h_{m+1,m}\boldsymbol{e}_m^H \exp(-tH_m)\boldsymbol{e}_1; \tag{2.5}$$

see [38, 40, 46, 61]. According to (2.5), the residual vectors can be represented as a time-dependent *scalar* function $\beta_m(t)$ times the next Krylov basis vector, and all quantities necessary for evaluating $\beta_m(t)$ are readily available from the Krylov iteration. Due to the unit norm of $\boldsymbol{v}_{m+1}$, we clearly have

$$\|\boldsymbol{r}_m(t)\| = |\beta_m(t)|.$$

Furthermore, one can show that for *any* $m \geq 1$ and *any* tolerance ε, there exists an interval $[0, t_0]$ such that

$$|\beta_m(s)| \leq \varepsilon \quad \text{for all } s \in [0, t_0];$$

see [40, Lemma 1] for a more precise statement. Thus, in order to find a suitable step size t_0 for performing a “restart” as outlined above, it is proposed in [40, Section 2.3] to monitor $\beta_m(\delta_i)$ at suitably chosen sampling points δ_i in $[0, 1]$. Each evaluation of β_m requires computing the exponential of an $m \times m$ Hessenberg matrix, which therefore has about the same cost as forming the coefficient vector of the Arnoldi approximation after m iterations. Thus, if $\mathcal{O}(1)$ sampling points are used for finding t_0, the asymptotic cost of the algorithm stays the same (and the additional amount of work is hardly noticeable for large scale problems). We briefly mention that further optimizations are possible: In [40, Section 2.4], the authors describe an adaptive version of the restarting procedure which also tries to adapt the Krylov dimension m from one cycle to the next in order to improve the overall run time.[13]

An extension of the RT restarting methodology to shift-invert [40, Section 2.5] or more general rational Krylov methods is also straightforwardly possible, although restarts typically play a less important role there due to the much smaller subspace dimensions.

In addition to the exponential function, RT restarting can also be applied to a few other matrix functions: In [41], the authors consider the inhomogeneous IVP

$$\boldsymbol{u}'(t) = -A\boldsymbol{u}(t) + \boldsymbol{g}, \qquad \boldsymbol{u}(0) = \boldsymbol{b},$$

which is solved by

$$\boldsymbol{u}(t) = \boldsymbol{b} + t\varphi(-tA)(\boldsymbol{g} - A\boldsymbol{b}),$$

where φ is defined in (1.4).

In our paper [S4] we further extend the methodology to certain second-order ODEs, whose solution involves trigonometric matrix functions and matrix square roots; see Section 2.1.3.

We conclude this section by mentioning a rather simplistic—but often quite efficient—alternative to restarting in the Hermitian case (where available memory is the only limiting factor): In order to avoid storing the complete Krylov basis V_m, one employs a *two pass* method [37,85]. In a first pass, one performs the Lanczos process for assembling the Hessenberg matrix H_m, directly discarding a basis vector once it is not needed any longer (so that only three basis vectors need to be kept in memory at a time). After this, the coefficient vector $\boldsymbol{c} := \|\boldsymbol{b}\| f(H_m)\boldsymbol{e}_1$ is formed and a second run of the Lanczos method is started, in which the basis is built again and $\boldsymbol{f}_m = \sum_{i=1}^{m} \boldsymbol{c}(i)\boldsymbol{v}_i$ is accumulated on the fly, again allowing to discard “old” basis vectors.

This approach essentially doubles the computational cost[14] but in principle allows to perform an arbitrary number of Krylov iterations, irrespective of the available memory.[15] This is the method that all restarted approaches need to be competitive with in the Hermitian case.

[13] In the Hermitian case—ignoring some academic examples—it is in general advisable to choose the restart length as large as the available memory permits, as this tends to lead to the overall smallest number of iterations. In the non-Hermitian case, where orthogonalization cost is also an issue, the “optimal” restart length might be much smaller than the maximum admissible length, though.

[14] At least in terms of matrix-vector products: The second pass does not require further inner products, as these are already stored in H_m.

[15] Note, however, that if $m = \mathcal{O}(1000)$, evaluating $f(H_m)$ can be become non-negligible.

2.1.1 A. Frommer, K. Kahl, M. Schweitzer, and M. Tsolakis, *Krylov subspace restarting for matrix Laplace transforms*, SIAM J. Matrix Anal. Appl., 44 (2023)

In the algorithm-oriented paper [S5], we extended the quadrature-based error function restarting approach from [81] to functions f which are Laplace transforms, cf. Definition 1.9.[16] When f is a Laplace transform,

$$f(A)\boldsymbol{b} = \int_0^\infty \exp(-tA)w(t)\,\mathrm{d}t\ \boldsymbol{b},$$

and it is this connection to the matrix exponential that can be exploited for finding an error function representation for Krylov approximations of $f(A)\boldsymbol{b}$. The basis of this derivation is a well-known error representation for the exponential: The difference of $\exp(-tA)\boldsymbol{b}$ and its Krylov approximation can be written as

$$\begin{aligned}\exp(-tA)\boldsymbol{b} &- \|\boldsymbol{b}\|V_m\exp(-tH_m)\boldsymbol{e}_1 \\ &= -h_{m+1,m}\|\boldsymbol{b}\|\int_0^t \exp((\tau-t)A)g(\tau)\,\mathrm{d}\tau\ \boldsymbol{v}_{m+1}\end{aligned} \tag{2.6}$$

where

$$g(\tau) = \boldsymbol{e}_m^H\exp(-\tau H_m)\boldsymbol{e}_1 \tag{2.7}$$

is the $(m,1)$ entry of $\exp(-\tau H_m)$; see, e.g., [166, Theorem 3.1]. By integrating over all errors (2.6) for t ranging from 0 to ∞, one obtains a representation of the error of the Krylov approximation of $f(A)\boldsymbol{b}$ as a Laplace transform of a different measure [S5, Theorem 3.2]. To be specific,

$$f(A)\boldsymbol{b} - \|\boldsymbol{b}\|V_mf(H_m)\boldsymbol{e}_1 = -h_{m+1,m}\|\boldsymbol{b}\|\int_0^\infty \exp(-tA)\widetilde{w}(t)\,\mathrm{d}t\ \boldsymbol{v}_{m+1},$$

where

$$\widetilde{w}(t) = \int_0^\infty w(t+\tau)g(\tau)\,\mathrm{d}\tau$$

and $g(\tau)$ is defined in (2.7).[17]

The result of [S5, Theorem 3.2] discussed above corresponds to the situation after the first cycle of a restarted method. To allow for an arbitrary number of restarts, this result needs to be applied recursively. Denoting by $\boldsymbol{f}_m^{(k)}$ the Arnoldi approximation at the end of the kth restart cycle, $k \geq 1$, we have

$$f(A)\boldsymbol{b} - \boldsymbol{f}_m^{(k)} = (-1)^k\left(\prod_{j=1}^k h_{m+1,m}^{(j)}\right)\|\boldsymbol{b}\|\int_0^\infty \exp(-tA)w^{(k+1)}(t)\,\mathrm{d}t\ \boldsymbol{v}_{m+1}^{(k)}, \tag{2.8}$$

[16] For convenience we restrict to the case of “standard” Laplace transforms, where the measure takes the form $\mathrm{d}\sigma(t) = w(t)\,\mathrm{d}t$.

[17] Guaranteeing the existence of this integral requires nontrivial considerations about the region of absolute convergence of the new Laplace transform; see [S5, Appendix A].

where

$$w^{(j)}(t) = \int_0^\infty w^{(j-1)}(t+\tau) g^{(j-1)}(\tau)\,\mathrm{d}\tau, \quad j \geq 2,$$

with $w^{(1)} = w$ and $g^{(j)}(\tau) = \boldsymbol{e}_m^H \exp(-\tau H_m^{(j)})\boldsymbol{e}_1$ for $j \geq 1$, see [S5, Corollary 3.3].

In principle, the representation (2.8) can be used as the basis of a restarted Krylov method for $f(A)\boldsymbol{b}$. From a practical perspective, it has the severe disadvantage that in the kth restart cycle, $w^{(k)}$ is defined via a k-fold iterated integral, which is not only cumbersome to implement but also leads to an exponential growth of the computational cost in the number of restart cycles when the integrals need to be approximated by numerical quadrature. As a means to overcome this problem, we propose in [S5, Section 5.2] to replace $w^{(j)}$ in (2.8) by an interpolating cubic spline $s^{(j)}$, which can be evaluated at arbitrary points without needing to perform any further costly computations with quantities from earlier cycles. This spline can be efficiently updated from one cycle to the next and the knots used for constructing it can be adaptively refined until the desired accuracy is reached.

As (outer) quadrature rule for (2.8), we use an adaptive Gauss–Kronrod (G7-K15) quadrature scheme [52] applied to a subdivision of $[0, 1)$ after applying the variable transformation $x = \sqrt{t}/(1 + \sqrt{t})$.

The resulting algorithm is tested in a number of numerical experiments that show that it is both numerically stable and can outperform competing algorithms. As an example, we summarize (a part of) the experiment in [S5, Section 6.1], where the approximation of $f(z) = z^{-3/2}$, the Laplace transform of $w(t) = \sqrt{t}$, is considered. As a method to compare against, we use `funm_quad`, our implementation of the restarted algorithm from [81]. While f is not a Stieltjes function, it can be written as $f(z) = z^{-1/2} \cdot z^{-1}$. Thus, $f(A)\boldsymbol{b}$ can be approximated by first (approximately) solving a linear system[18] to obtain $A^{-1}\boldsymbol{b}$ and then applying a restarted Arnoldi method for the inverse square root to the resulting vector.

We test our algorithm for a discretization of the three-dimensional Laplace operator and for a discretization of a three-dimensional convection-diffusion equation (with constant convection field), both on an equispaced grid, resulting in a Hermitian and a non-Hermitian model problem. Figure 2.3 reports the run time of the two algorithms (and, for the Hermitian problem, of the two-pass Lanczos method) for various grid sizes N. While our method is at least on par with two-pass Lanczos in the Hermitian case, it excels in the non-Hermitian case, showing a better scaling behavior and run times that are almost a factor 3 smaller for the largest considered problem. For details about the algorithmic setup, parameter choices etc., we refer to [S5, Section 6.1].

[18] In our experiments, we use the MATLAB implementations of the conjugate gradient method [93] and restarted GMRES [145] for this, depending on whether A is Hermitian positive definite or not.

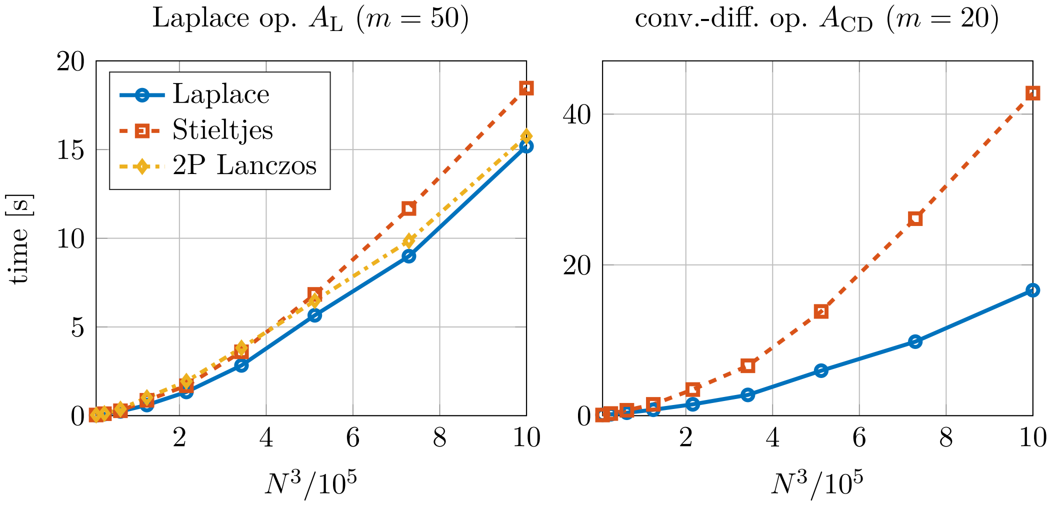


Figure 2.3: Execution times when approximating $A^{-3/2}b$ for varying matrix sizes $n = N^3$. "Laplace" denotes the algorithm from [S5], "Stieltjes" is the combination of `funm_quad` with CG (left) or GMRES (right). "2P Lanczos" denotes the two-pass Lanczos method. The restart length is m. *Originally [S5, Figure 6.4].*

As additional contributions, the paper [S5] also discusses the extension of the restart methodology to two-sided Laplace transforms and to Bernstein functions, thus widening the scope of functions to which it can be applied in a stable fashion.[19] We also worked out the precise connection to error function restarting for Stieltjes functions—which are a special case of Laplace transforms, see Section 1.3—in [S5, Corollary 3.5].

We mention in passing that subsequently, convergence of restarted Arnoldi for Laplace transforms was further investigated in [165, Section 4.4–4.5].

2.1.2 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)

Our paper [S9] provides a unified view on convergence theory for (Krylov) methods that can be applied to compute the action of functions of Hermitian matrices in the large-scale setting, when only limited memory is available. In particular, we compare the restarted Arnoldi method [65, 81, 160], the multi-shift CG algorithm applied to a rational approximation $r \approx f$ [68, 83], the two-pass Lanczos

[19] Notable examples of functions that belong to these classes are the square root and the gamma function.

method [85], as well as rational Krylov methods with polynomial inner solves (cf. Section 1.4). The latter can be considered as limited memory methods because the number of outer iterations will likely be small, while for the inner iteration, a short-recurrence method (like the conjugate gradient method) can be used.

The overall goal of [S9] is to answer the following question, using only information that is available a priori (or easy to compute):

"Given a limited amount of memory (storage of at most $m_{\max}$ vectors of length n) and a target accuracy ε, what is an efficient way to extract an accurate approximation to $f(A)\boldsymbol{b}$ from a polynomial Krylov space?"

Our analysis focuses on the class of (Cauchy–)Stieltjes functions because they are rather well-understood from a theoretical perspective and we were able to, e.g., use the convergence results obtained in [80] in our analysis.

The first part of the paper has a survey-like character, reviewing the different methods and their convergence theory and putting them into a unified framework. In this part, we also derive a new convergence estimate for the shift-invert Lanczos method that is similar in spirit (and proof technique) to its polynomial counterpart in [80].

In order to be able to make meaningful predictions and obtain a fair comparison, the second part of the paper deals with inexactness in rational Krylov methods, specifically in the shift-invert method.[20] It is well-known in the context of inexact Krylov methods for linear systems that the allowed inexactness can gradually increase as the main iteration converges, without spoiling the overall convergence of the method [42, 70, 156], so that later iterations become much cheaper.

In [S9, Section 4 & Appendix A], we derive similar results for the shift-invert method. Denoting again $B = (A - \xi I)^{-1}$, the starting point of our derivation is the inexact Arnoldi decomposition

$$B(V_m - R_m) = V_m H_m + h_{m+1,m}\boldsymbol{v}_{m+1}\boldsymbol{e}_m^H,$$

where the columns $\boldsymbol{r}_j$ of R_m are the residuals incurred when approximately solving $(A - \xi I)\boldsymbol{w} = \boldsymbol{v}_j$. This can be rewritten as

$$(B + E_m)V_m = V_m H_m + h_{m+1,m}\boldsymbol{v}_{m+1}\boldsymbol{e}_m^H,$$

where $E_m := -BR_mV_m^H$ is a matrix of rank at most m. This can be interpreted as follows: The inexact shift-invert method computes an Arnoldi approximation $\boldsymbol{g}_m := \|\boldsymbol{b}\|V_m g(H_m)\boldsymbol{e}_1$ to $g(B + E_m)\boldsymbol{b}$,[21] instead of to the actually sought after

[20] The obtained results can be straightforwardly generalized to extended Krylov methods, though.

[21] We remind the reader that $g(y) := f(y^{-1} + \xi)$.

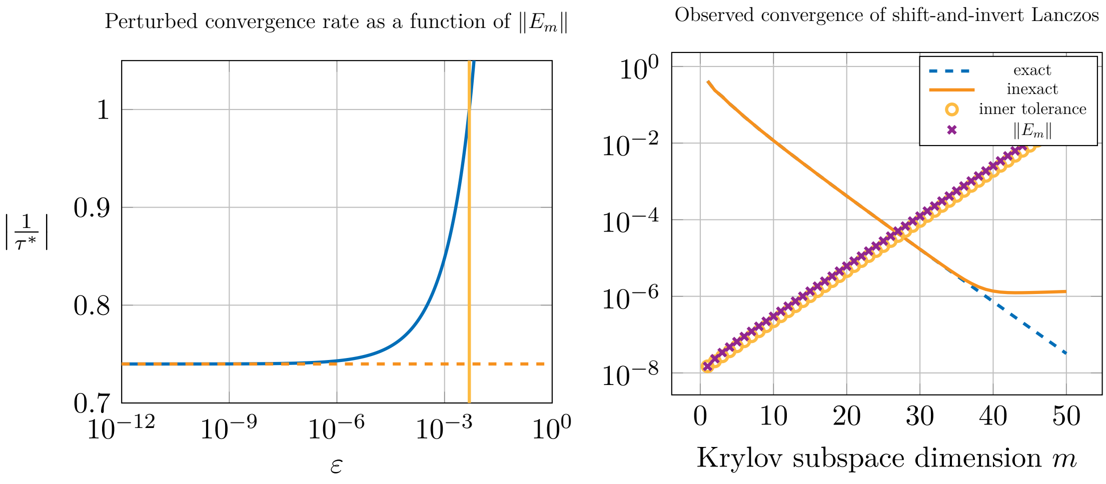


Figure 2.4: *Left:* Convergence rate for the perturbed problem as a function of $\|E_m\|$. The horizontal, dashed line shows the convergence rate of the unperturbed problem and the vertical dash-dotted line shows $(\lambda_{\max} - \xi)^{-1}$. *Right:* Convergence of the exact and inexact shift-invert method when aiming for an overall tolerance of $\texttt{tol} = 10^{-6}$ and using the relaxation strategy outlined in (2.9)–(2.10). In both cases, A is a diagonal matrix with Chebyshev eigenvalues in $[0.1, 200]$. *Originally [S9, Figure A.2].*

$g(B)\boldsymbol{b} = f(A)\boldsymbol{b}$. An estimate of the distance between the shift-invert method and $f(A)\boldsymbol{b}$ can thus be obtained as

$$\|f(A)\boldsymbol{b} - \boldsymbol{g}_m\| \le \|g(B + E_m)\boldsymbol{b} - \boldsymbol{g}_m\| + \|g(B)\boldsymbol{b} - g(B + E_m)\boldsymbol{b}\|,$$

where the first term measures the Arnoldi error and the second term measures the inexactness in the quantity that is computed.

By carefully analyzing both terms individually and exploiting geometric decay in the coefficients of the Krylov solution, we find the following relaxation strategy, which turns out to work well in practice: When aiming for an overall accuracy ε, solve the first inner linear system to a residual norm of

$$\varepsilon_1 := \frac{\varepsilon}{2\big((\lambda_{\min} - \xi)|f'(\lambda_{\min})| + (\lambda_{\max} - \xi)|f'(\sqrt{\lambda_{\min}\lambda_{\max}})|\big)} \tag{2.9}$$

and then let the residual norm grow geometrically in further iterations,

$$\varepsilon_j = \frac{\varepsilon_1}{\alpha^{j-1}}, j = 2, 3, \ldots, \qquad \text{where} \qquad \alpha = \frac{\sqrt[4]{\kappa(A)} - 1}{\sqrt[4]{\kappa(A)} + 1} \tag{2.10}$$

is the convergence factor from an a priori convergence bound for shift-invert Lanczos (with an optimized pole ξ).

Figure 2.4 illustrates the influence of this relaxation strategy on the convergence of the overall method for a simple model problem: We take $A \in \mathbb{C}^{1{,}000\times 1{,}000}$ diagonal with Chebyshev eigenvalues in $[0.1, 200.1]$ and $\boldsymbol{b}$ a (normalized) vector of all ones and aim to approximate $A^{-1/2}\boldsymbol{b}$ to a relative accuracy of 10^{-6}.

In a series of numerical experiments, we confirm our theoretical comparison and illustrate that the a priori convergence estimates can often successfully be used to correctly predict the *"relative performance"* between the different considered methods, thus allowing to use them for selecting the most appropriate method for a given setting. On the other hand, predicting the actual *number* of matrix-vector products that is necessary for convergence to a given accuracy turns out to be very difficult. This comes as no surprise, as the underlying convergence estimates typically involve large constants that are mostly artifacts of the proof technique but are not descriptive of the actual behavior of the method.

Overall, the restarted Lanczos method and the multishift CG method (if a good rational approximation $r \approx f$ is readily available) turned out to be the methods with the best combination of ease of use, reliability and efficiency.

2.1.3 M. A. Botchev, L. A. Knizhnerman, and M. Schweitzer, *Krylov subspace residual and restarting for certain second order differential equations*, SIAM J. Sci. Comput., 46 (2024)

In [S4], we consider an extension of RT restarting to second-order IVPs

$$\boldsymbol{y}''(t) = -A\boldsymbol{y}(t) + \boldsymbol{g}, \quad \boldsymbol{y}(0) = \boldsymbol{u}, \quad \boldsymbol{y}'(0) = \boldsymbol{v}, \tag{2.11}$$

which are solved by

$$\boldsymbol{y}(t) = \boldsymbol{u} + \frac{1}{2}t^2\psi(t^2A)(-A\boldsymbol{u} + \boldsymbol{g}) + t\sigma(t^2A)\boldsymbol{v},$$

where ψ and σ are the entire functions

$$\psi(x^2) = 2\frac{1-\cos x}{x^2}, \quad \sigma(x^2) = \frac{\sin x}{x},$$

where we set $\psi(0) = 1$ and $\sigma(0) = 1$. Clearly (2.11) can be transformed into a first-order ODE system

$$\boldsymbol{w}'(t) = -\mathcal{A}\boldsymbol{w}(t) + \widehat{\boldsymbol{g}}, \quad \boldsymbol{w}(0) = \boldsymbol{w}_0,$$

where

$$\boldsymbol{w}(t) = \begin{bmatrix} \boldsymbol{y}(t) \\ \boldsymbol{y}'(t) \end{bmatrix}, \quad \boldsymbol{w}_0 = \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{bmatrix}, \quad \widehat{\boldsymbol{g}} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{g} \end{bmatrix}, \quad \mathcal{A} = \begin{bmatrix} 0 & -I \\ A & 0 \end{bmatrix},$$

to which the RT restarting approach from [40,41] could directly be applied. From a computational perspective, this has several drawbacks, though. Even if A is Hermitian, the matrix $\mathcal{A}$ is not, and additionally, this approach requires working with vectors twice the size of the original problem, which increases orthogonalization cost and storage requirements. It is therefore desirable to be able to directly work with the second-order formulation (2.11) instead.

To solve (2.11) by a Krylov method, one needs to build bases of *two* Krylov spaces, $\mathcal{K}_m(A, -A\boldsymbol{u}+\boldsymbol{g})$ and $\mathcal{K}_m(A, \boldsymbol{v})$, yielding the associated Arnoldi decompositions

$$\begin{aligned} AV_m^{(\psi)} &= V_m^{(\psi)} H_m^{(\psi)} + h_{m+1,m}^{(\psi)} \boldsymbol{v}_m^{(\psi)} \boldsymbol{e}_m^H \\ AV_m^{(\sigma)} &= V_m^{(\sigma)} H_m^{(\sigma)} + h_{m+1,m}^{(\sigma)} \boldsymbol{v}_m^{(\sigma)} \boldsymbol{e}_m^H. \end{aligned}$$

An approximate Krylov solution is then obtained as

$$\boldsymbol{y}_m(t) = \boldsymbol{u} + \boldsymbol{y}_m^{(\psi)}(t) + \boldsymbol{y}_m^{(\sigma)}(t), \tag{2.12}$$

with

$$\boldsymbol{y}_m^{(\psi)}(t) = \frac{1}{2}\|-A\boldsymbol{u}+\boldsymbol{g}\| t^2 V_m^{(\psi)} \psi(t^2 H_m^{(\psi)})\boldsymbol{e}_1, \quad \boldsymbol{y}_m^{(\sigma)}(t) = \|\boldsymbol{v}\| V_m^{(\sigma)} \sigma(t^2 H_m^{(\sigma)})\boldsymbol{e}_1.$$

One can then show that the residual

$$\boldsymbol{r}_m(t) = -\boldsymbol{y}_m''(t) - A\boldsymbol{y}(t) + \boldsymbol{g}$$

of the approximation (2.12) fulfills a relation that is similar to the first-order case, namely

$$\boldsymbol{r}_m(t) = \boldsymbol{r}_m^{(\psi)}(t) + \boldsymbol{r}_m^{(\sigma)}(t),$$

with

$$\begin{aligned} \boldsymbol{r}_m^{(\psi)}(t) &= -\beta_m^{(\psi)}(t)\boldsymbol{v}_{m+1}^{(\psi)}, \qquad & \beta_m^{(\psi)}(t) = \frac{1}{2}\|-A\boldsymbol{u}+\boldsymbol{g}\| t^2 h_{m+1,m}^{(\psi)} \boldsymbol{e}_m^T \psi(t^2 H_m^{(\psi)})\boldsymbol{e}_1, \\ \boldsymbol{r}_m^{(\sigma)}(t) &= -\beta_m^{(\sigma)}(t)\boldsymbol{v}_{m+1}^{(\sigma)}, \qquad & \beta_m^{(\sigma)}(t) = \frac{1}{2}\|\boldsymbol{v}\| h_{m+1,m}^{(\sigma)} \boldsymbol{e}_m^T \sigma(t^2 H_m^{(\sigma)})\boldsymbol{e}_1. \end{aligned}$$

Based on the above representation of the residual, one can show that for any $m \geq 1$, the residual norm becomes arbitrarily small if t is chosen small enough. In particular,

$$\|\boldsymbol{r}_m(t)\| \leq t\varphi(-t\widehat{\omega}) \left(h_{m+1,m}^{(\psi)} \|-A\boldsymbol{u}+\boldsymbol{g}\| + h_{m+1,m}^{(\sigma)} \|\boldsymbol{v}\| \right), \tag{2.13}$$

where $\varphi(z)$ is defined in (1.4) and

$$\widehat{\omega} = \min \left\{ -\frac{1}{2}\|H_m^{(\psi)} - I\|, -\frac{1}{2}\|H_m^{(\sigma)} - I\| \right\};$$

see [S4, Proposition 2.2]. Better bounds can be obtained by expanding the residual in terms of Chebyshev or Faber polynomials, as we show in [S4, Section 3], but we do not report this rather technical analysis here.

Equation (2.13) directly implies that for any $\varepsilon > 0$ there exists $t_0 > 0$ such that $\|\boldsymbol{r}_m(s)\| \leq \varepsilon$ for $s \in [0, t_0]$, and a RT restarting scheme can be built upon this observation, just as in the first-order case. There are several nontrivial implementation issues to consider which do not occur in the first-order setting, though. Many of these are related to the fact that *two* Krylov spaces need to be built in order to approximate the solution of (2.11). Therefore, in order to form the approximation (2.12) after determining a suitable step size, two Krylov bases need to be kept in memory at the same time, i.e., if a total of m basis vectors can be stored, then only $m/2$ iterations can be performed for each Krylov space, which generally slows down convergence. If, on the other hand, the Krylov spaces are built sequentially, one after the other, selecting an appropriate time step and imposing a suitable residual tolerance becomes more cumbersome, as not all relevant quantities are available at the same time. In [S4, Section 2.4], we discuss all these issues and present several different algorithmic variants of the RT restarting method, tailored to different possible scenarios.

As a possibility to speed up Krylov methods for (2.11) within the RT restarting framework, we discuss a combination with the *Gautschi cosine scheme* [39,86,100] in [S4, Section 2.5]. This scheme is based on the observation that, for any time step δ, the solution of (2.11) satisfies

$$\boldsymbol{y}(t+\delta) - 2\boldsymbol{y}(t) + \boldsymbol{y}(t-\delta) = \delta^2\psi(\delta^2 A)(-A\boldsymbol{y}(t) + \boldsymbol{g}), \tag{2.14}$$

which follows from the variation-of-constants formula. When using a fixed step size δ and denoting $\boldsymbol{y}_k = \boldsymbol{y}(k\delta)$, by straight-forward algebraic manipulations, equation (2.14) can be recast as the time stepping scheme

$$\begin{aligned} \boldsymbol{v}_{k+1/2} &= \boldsymbol{v}_k + \frac{1}{2}\delta\psi(\delta^2 A)(-A\boldsymbol{y}_k + \boldsymbol{g}), \\ \boldsymbol{y}_{k+1} &= \boldsymbol{y}_k + \delta\boldsymbol{v}_{k+1/2}, \\ \boldsymbol{v}_{k+1} &= \boldsymbol{v}_{k+1/2} + \frac{1}{2}\delta\psi(\delta^2 A)(-A\boldsymbol{y}_{k+1} + \boldsymbol{g}), \end{aligned} \tag{2.15}$$

with starting vectors $\boldsymbol{y}_0 := \boldsymbol{u}$ and $\boldsymbol{v}_0 := \sigma(\delta^2 A)\boldsymbol{v}$.

Using the scheme (2.15) requires evaluating the σ-function just once for forming the starting vector $\boldsymbol{v}_0$ and evaluating the ψ-function once per iteration,[22] thus effectively cutting the computational cost in half and avoiding the necessity to build two different Krylov spaces per cycle. The difficulty in efficiently employing the Gautschi scheme in practice lies in the fact that the step size δ needs to be fixed once and for all at the start of the iteration.

[22] Note that the second evaluation of ψ in (2.15) coincides with the first evaluation in the next iteration.

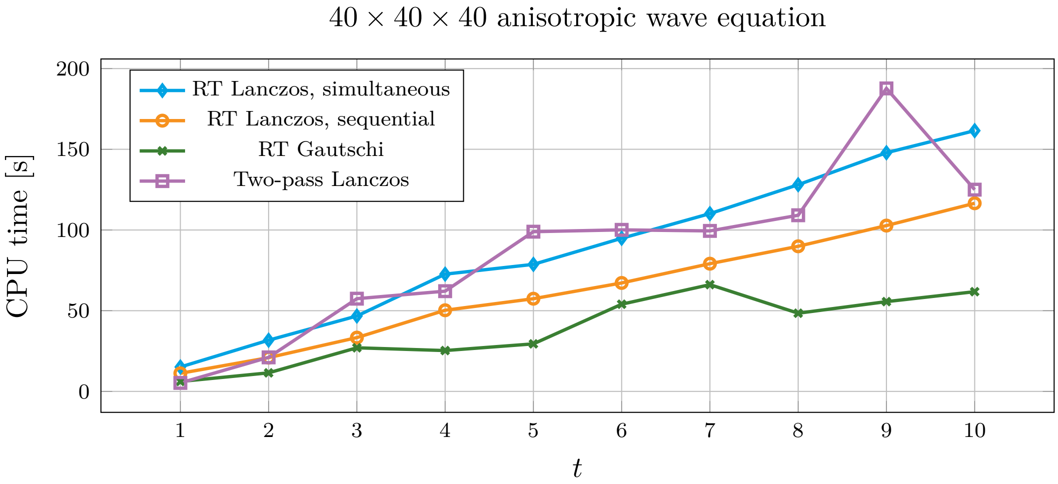


Figure 2.5: CPU time used by the different algorithms for solving the anisotropic wave equation, depending on the final time t. *Originally [S4, Figure 3].*

Based on the maximum admissible Krylov dimension m and the desired accuracy, one can determine a suitable δ in the first iteration (or rather when forming $\boldsymbol{v}_0$) based on the residual norm. However, there is no guarantee that the desired accuracy can also be reached within m Krylov steps in later cycles when using this δ. Thus, one might be tempted to use a conservative step size selection and choose δ much smaller than what the residual norm in the first cycle suggests, which of course increases the overall cost of the method. This is where residual restarting comes into play. Should it turn out that it is not possible to compute $\boldsymbol{v}_{k+1}$ to the desired accuracy in a later cycle because the time step δ is too large, one can perform a "mini restart" to bridge the gap: Based on the residual norm, one finds $\widetilde{\delta}$ for which the approximation is accurate enough and then restarts to complete the step from $\widetilde{\delta}$ to δ (which typically only requires very few additional Krylov iterations). This way, one can continue the iteration, instead of needing to start from scratch with a smaller δ.

As an illustration of the performance of the different algorithmic variants, we consider the solution of the anisotropic three-dimensional wave equation

$$\begin{cases} u_{tt} = k_x u_{xx} + k_y u_{yy} + k_z u_{zz}, \\ u(0,x,y,z) = u_0(x,y,z), \quad u_t(0,x,y,z) = v_0(x,y,z), \end{cases} \tag{2.16}$$

on the domain $\Omega = (0,1) \times (0,1) \times (0,1)$, with coefficients $k_x = 10^4, k_y = 10^2, k_z = 1$ and initial conditions $u_0(x,y,z) = \sum_{i,j,k=1}^{3} \sin(i\pi x)\sin(j\pi y)\sin(k\pi z), v_0(x,y,z) = \sum_{i,j,k=1}^{3} \lambda_{ijk} \cdot \sin(i\pi x)\sin(j\pi y)$, where $\lambda_{ijk} = \pi^2(i^2k_x + j^2k_y + k^2k_z)$.

Figure 2.5 shows the run time that the different RT restarting schemes (with $m = 30$) require to solve a semi-discretization of (2.16) on a 40 × 40 × 40 grid to an accuracy of 10^{-6} for different final times t, together with the run time of the two-pass Lanczos method as baseline. It is clearly visible that for larger t, i.e., more difficult problems, the sequential RT restarting scheme and RT Gautschi scheme outperform the two-pass method. This is mostly due to the fact that very large Krylov dimensions are required for such large time steps, so that solving the compressed problem in two-pass Lanczos becomes non-negligible.[23]

2.2 Krylov methods with randomized sketching

A rather new trend in numerical linear algebra is the adoption of fast and scalable randomized algorithms for many core problems. The most prominent example of this is probably the randomized singular value decomposition [92], which is widely used today.

An important technique that has emerged in randomized numerical linear algebra is the *sketch-and-solve* paradigm. It was initially designed and used for computations with rather crude accuracy demands, as for example (low-rank) matrix approximation [146, 171] or construction of preconditioners [9], but quite recently—starting with [11, 133]—it was discovered that it might also be a viable technique for high accuracy computations when appropriately combined with Krylov subspace methods.

At the heart of sketching algorithms are so-called *subspace embeddings* [60, 146, 170], which allow to embed a (low-dimensional) subspace $\mathcal{V}$ of $\mathbb{R}^n$ into a Euclidean space $\mathbb{R}^s$ of smaller dimension $s \ll n$, such that norms are distorted in a controlled manner. To be more precise, for given $\varepsilon \in [0, 1)$, a matrix $S \in \mathbb{R}^{s\times n}$ is called ε-subspace embedding if

$$(1-\varepsilon)\|\boldsymbol{v}\|^2 \leq \|S\boldsymbol{v}\|^2 \leq (1+\varepsilon)\|\boldsymbol{v}\|^2, \tag{2.17}$$

for all $\boldsymbol{v} \in \mathcal{V}$, or equivalently,

$$\langle \boldsymbol{u}, \boldsymbol{v}\rangle - \varepsilon\|\boldsymbol{u}\|\|\boldsymbol{v}\| \leq \langle S\boldsymbol{u}, S\boldsymbol{v}\rangle \leq \langle \boldsymbol{u}, \boldsymbol{v}\rangle + \varepsilon\|\boldsymbol{u}\|\|\boldsymbol{v}\|$$

for all $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{V}$.

Such an embedding is called *oblivious* if it can be constructed without explicit knowledge of $\mathcal{V}$. This is, e.g., relevant in the context of Krylov methods, as the final Krylov subspace is not known when starting the method. Oblivious embeddings can be constructed by probabilistic methods which only require $m = \dim(\mathcal{V})$ and the target dimension s of the embedding space as input and then

[23] See [S4, Section 4.1] for a detailed discussion of parameters and implementation aspects, as well as the unusual spike in the two-pass run time for $t = 9$.

construct S such that (2.17) is fulfilled with high probability. These constructions often involve randomized subsampled trigonometric transforms; see, e.g., [125, 164]. As the required sketching dimension typically depends quadratically on the distortion ε, one needs to work with rather crude accuracies, with $\varepsilon = 1/\sqrt{2}$ or $\varepsilon = 1/2$ being common choices.

An alternative viewpoint which is sometimes useful is that a subspace embedding S induces a semidefinite inner product

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle_S := \langle S\boldsymbol{u}, S\boldsymbol{v} \rangle, \tag{2.18}$$

which, when restricted to $\mathcal{V}$, is an actual (positive definite) inner product; see, e.g., [12, Section 3.1].

The combination of Krylov methods with randomized sketching comes in two "flavors": The first class of methods performs the usual Arnoldi process, but uses a Gram–Schmidt orthogonalization with respect to the semidefinite inner product (2.18) instead of the standard Euclidean inner product. Examples of this methodology include [10, 11, 163] for linear systems and eigenvalue problems and [48, Algorithm 3] for matrix functions. The second flavor of sketched Krylov methods employs a truncated Arnoldi method (which produces a non-orthogonal basis) and uses randomized techniques for mitigating the negative effects of lack of orthogonality. Notable work in this direction includes the seminal paper [133] covering linear systems and eigenvalue problems, and our own work [S10] which is described in more detail in Section 2.2.1, as well as [48, Algorithm 4] for the matrix function case.

2.2.1 S. Güttel and M. Schweitzer, *Randomized sketching for Krylov approximations of large-scale matrix functions*, SIAM J. Matrix Anal. Appl., 44 (2023)

The main idea of our paper [S10] was to transfer the techniques introduced in [133] for linear systems to matrix functions in order to reduce computational complexity and communication (and possibly also storage demands) in Krylov subspace methods. The starting point for doing so is the use of any computationally cheap method that produces a *non-orthogonal* Krylov basis W_m of $\mathcal{K}_m(A, \boldsymbol{b})$. In our experiments, using a k-truncated Arnoldi process[24] for this purpose—which reduces the orthogonalization cost from $\mathcal{O}(m^2 n)$ to $\mathcal{O}(mnk)$—turned out to be most successful.

[24] By this we mean an Arnoldi process which orthogonalizes a new basis vector only against the k previous basis vectors, which, e.g., for $k = 2$ mimics the Hermitian Lanczos process.

Working with the non-orthogonal W_m, one could in principle still obtain the Arnoldi approximation $\boldsymbol{f}_m$ via (1.12). However, as this formula involves the Moore-Penrose pseudoinverse, both its computational cost and numerical stability properties are not very attractive. To overcome this, we propose an alternative extraction procedure, which is easiest to explain for the linear system case. Assuming we want to obtain an approximate solution $\widehat{\boldsymbol{x}}_m(z)$ for a shifted linear system $(zI - A)\boldsymbol{x}(z) = \boldsymbol{b}$ from $\mathcal{K}_m(A, \boldsymbol{b})$, we demand a *sketched Galerkin condition*[25]

$$\widehat{\boldsymbol{x}}_m(z) = W_m \widehat{\boldsymbol{y}}_m(z) \quad \text{with} \quad (SW_m)^H[S\boldsymbol{b} - S(zI - A)\widehat{\boldsymbol{x}}_m(z)] = \boldsymbol{0}. \tag{2.19}$$

If the inverted quantity in the next equation is well-defined, the coefficient vector $\widehat{\boldsymbol{y}}_m(t)$ can equivalently be written as

$$\widehat{\boldsymbol{y}}_m(z) = [(SW_m)^H(zSW_m - SAW_m)]^{-1}(SW_m)^H(S\boldsymbol{b}).$$

A corresponding approximation for $f(A)\boldsymbol{b}$ from $\mathcal{K}_m(A, \boldsymbol{b})$ can be obtained by integrating (2.19) along a suitable contour Γ, yielding after straightforward algebraic manipulations

$$\begin{aligned} \widehat{\boldsymbol{f}}_m &= \frac{1}{2\pi i}\int_\Gamma f(z) W_m[zW_m^H S^H S W_m - W_m^H S^H S A W_m]^{-1} \,\mathrm{d}z \; (SW_m)^H(S\boldsymbol{b}) \\ &= W_m(W_m^H S^H S W_m)^{-1} f\left(W_m^H S^H S A W_m (W_m^H S^H S W_m)^{-1}\right)(SW_m)^H(S\boldsymbol{b}). \end{aligned}$$

This *sketched FOM (sFOM)* approximation turned out to be numerically unstable in certain situations, but fortunately there is an easy remedy to this, the so-called *"basis whitening"* idea. To whiten the nonorthogonal Krylov basis W_m, one starts by computing a thin QR decomposition $SW_m = Q_m R_m$ of the *sketched* basis and then performs the replacements

$$SW_m \leftarrow Q_m, \quad SAW_m \leftarrow (SAW_m)R_m^{-1}, \quad W_m \leftarrow W_m R_m^{-1},$$

where the last replacement is only done implicitly: Writing the sFOM approximation in the whitened basis yields

$$\widehat{\boldsymbol{f}}_m = W_m\big(R_m^{-1} f\left(Q_m^H S A W_m R_m^{-1}\right) Q_m^H S\boldsymbol{b}\big). \tag{2.20}$$

By suitably ordering the computations involved in forming $Q_m^H SAW_m R_m^{-1}$, one can then avoid forming $W_m R_m^{-1}$ explicitly.[26]

Performing m Krylov steps and computing the sFOM approximation (2.20) requires an overall computational cost of $\mathcal{O}(nm \log m + m^3)$ when truncated orthogonalization with $k = \mathcal{O}(1)$ is used and the sketching matrix S is constructed

[25] A similar construction was used in [12] in the context of model reduction for PDEs.

[26] Explicitly forming this matrix product would incur a cost of $\mathcal{O}(m^2 n)$ and would therefore asymptotically be as expensive as simply performing the full Gram–Schmidt orthogonalization.

via a randomly subsampled fast Fourier transform or discrete cosine transform; see [S10, Section 2.2] for details. Additionally, due to the use of a k-term recurrence for basis construction, this approach allows to use a two-pass approach for non-Hermitian matrices in case that storage requirements become a limiting factor; see [S10, Section 4.2].

While the sFOM approximation (2.20) often works very successfully in practice, it is quite difficult to analyze theoretically, e.g., because sketching might significantly alter spectral properties of A. In our recent preprint [137], we perform an analysis that explains the observed convergence behavior at least for entire functions like the exponential.

In addition, practical problems can occur if it happens that an eigenvalue of the sketched-and-projected matrix $Q_m^H SAW_m R_m^{-1}$ collides with a singularity of f.[27]

To overcome these problems, we also introduced a *"sketched GMRES (sGMRES)"* approximation in [S10, Section 3], which is defined via

$$\widetilde{\boldsymbol{f}}_m = \frac{1}{2\pi i} \int_\Gamma f(z)(zSW_m - SAW_m)^\dagger S\boldsymbol{b} \,\mathrm{d}z \tag{2.21}$$

and can be seen as a sketched counterpart of the *harmonic* Arnoldi approximation discussed in [98] and [80, Section 6]. In contrast to the sFOM approximation (2.20), there is no closed-form expression for the sGMRES approximation (2.21) and it therefore needs be evaluated by (adaptive) numerical quadrature; see [S10, Section 4.1]. We note that in recent work [45], a similar (but not equivalent) sketched GMRES-type approximant for $f(A)\boldsymbol{b}$ was proposed which indeed exhibits a closed-form expression.

Despite the influence that sketching might have on spectral properties of A, it is possible to prove that the approximation (2.21) converges to $f(A)\boldsymbol{b}$ whenever f is a Cauchy–Stieltjes function and A is positive real.[28] Indeed, according to [S10, Theorem 3.3], we have

$$\|f(A)\boldsymbol{b} - \widetilde{\boldsymbol{f}}_m\|_{A^HA} \leq C_1 C_\epsilon \|\boldsymbol{b}\| (\sin(\beta_0))^m,$$

where $C_\varepsilon = \sqrt{(1+\varepsilon)/(1-\varepsilon)}$ is a constant that depends on the embedding quality, while $C_1 = \|A\| f(\rho \|A\|^2)$ and $\beta_0 = \arccos(\delta/\|A\|)$ are constants that depend on spectral properties of A. Here, δ is the smallest real part of any element in $W(A)$ and ρ is the smallest real part of any element in $W(A^{-1})$.

In [S10, Section 5], we illustrate the performance of the proposed methods on a variety of examples from different applications, both to demonstrate that they are

[27] For the standard Arnoldi method, this cannot happen when all singularities of f lie outside the field of values $W(A)$.

[28] A matrix A is called positive real if $W(A)$ lies in the open right half plane.

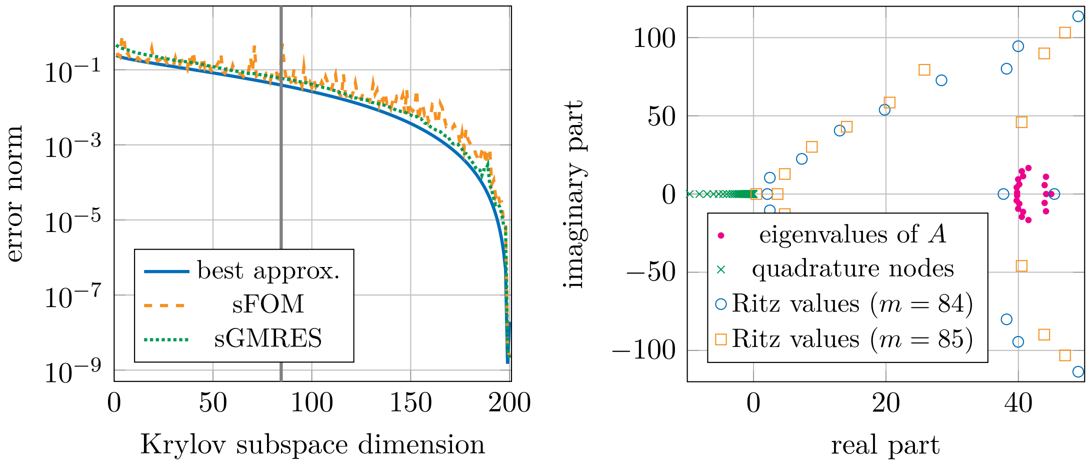


Figure 2.6: *Left:* Convergence of sketched Krylov methods based on truncated Arnoldi with truncation parameter $k = 4$ for the inverse square root of a convection diffusion operator. The error of best approximation to $A^{-1/2}\boldsymbol{b}$ from the Krylov space $\mathcal{K}_m(A, \boldsymbol{b})$ is also shown. *Right:* Sketched Ritz values $\mathrm{spec}(Q_m^H SAW_m R_m^{-1})$ for $m = 84$ and $m = 85$, which are closest to the gray vertical line at position $m = 84.5$ in the left plot. The jump in the sketched FOM error at $m = 85$ is caused by a Ritz value being very close to the origin. *Originally [S10, Figure 5.1].*

very competitive with state-of-the-art alternatives and to better understand their general behavior. We depict an exemplary convergence curve—which is quite typical for sketched methods—in Figure 2.6. It arises from approximating the inverse square root of a discretized convection diffusion operator with constant convection field. Overall, the performance of the methods closely resembles that of the standard Arnoldi method, with only a small offset in achieved error norm. While convergence of sGMRES is very smooth, the sFOM approximation often exhibits an erratic, “spikey” convergence curve, in particular in the initial phase of the method when convergence is slow. As the right part of the figure illustrates, spikes in the convergence curve are related to Ritz values coming close to the singularity at the origin (which is more likely to happen for the sketched method than for the standard method, as explained above).

CHAPTER 3

FRÉCHET DERIVATIVES AND LOW-RANK UPDATES OF MATRIX FUNCTIONS

In this chapter, we discuss both theoretical results and computational methods regarding the Fréchet derivative of matrix functions, as well as how to update a given, precomputed matrix function $f(A)$ to $f(A+E)$, where E is a matrix of low-rank. The common trait of these two at first sight quite different problems is that they both fit within the framework of *bivariate matrix functions*. Due to this connection—which will be made more precise below—they share many similarities both algorithmically and from a theoretical perspective.

In general, for an analytic, bivariate function $f : \mathbb{C}^2 \longrightarrow \mathbb{C}$ and two square matrices $A \in \mathbb{C}^{m\times m}, B \in \mathbb{C}^{n\times n}$ the bivariate matrix function $f\{A,B\}$ is a linear operator acting on $\mathbb{C}^{m\times n}$. One possible definition of $f\{A,B\}$, taken from [115], is given in the following.

Definition 3.1. Let $A \in \mathbb{C}^{m\times m}, B \in \mathbb{C}^{n\times n}$ and assume that there exist domains $\Omega_A, \Omega_B \subset \mathbb{C}$ such that $f_y(x) := f(x,y)$ is analytic on Ω_A for every $y \in \Omega_B$ and $f_x(y) := f(x,y)$ is analytic on Ω_B for every $x \in \Omega_A$. Then, the bivariate matrix function $f\{A,B\}$ is defined via

$$f\{A,B\}(E) := -\frac{1}{4\pi^2}\int_{\Gamma_A}\int_{\Gamma_B} f(x,y)(xI-A)^{-1}E(yI-B^H)^{-1}\,\mathrm{d}y\,\mathrm{d}x \quad (3.1)$$

for any $E \in \mathbb{C}^{m\times n}$, where $\Gamma_A \subset \Omega_A, \Gamma_B \subset \Omega$ are closed contours which wind around $\operatorname{spec}(A)$ and $\operatorname{spec}(B)$, respectively, in counterclockwise direction.

Besides the computation of Fréchet derivatives and low-rank updates covered in this chapter, further important examples of numerical linear algebra problems that can be rephrased as evaluating bivariate matrix functions are the solution of Sylvester and Stein matrix equations; see, e.g. [119, 155].

3.1 Fréchet derivatives of matrix functions

Given a matrix function f, its *Fréchet derivative* at the matrix A is an operator $L_f(A, \cdot)$ which is linear in the second argument and satisfies

$$f(A+E) - f(A) = L_f(A,E) + o(\|E\|), \quad \text{for all } E \in \mathbb{C}^{n\times n}.$$

The Fréchet derivative has many theoretical and practical applications. Most prominently, it plays an important role in computing or bounding the *condition number* of $f(A)$, via the relation

$$\operatorname{cond}(f,A) = \lim_{\varepsilon\to 0} \sup_{\|E\|\leq\varepsilon\|A\|} \frac{\|L_f(A,E)\|}{\|f(A)\|};$$

see, e.g., [94, Chapter 3] for details. Further applications which require the evaluation of $L_f(A,E)$ include analysis of complex networks [74], decomposition of tensor grids [104] or the solution of optimization problems involving matrix functions [162].

The Fréchet derivative fits into the framework of bivariate matrix functions as follows: Given a differentiable, univariate matrix function f, define the bivariate function[29]

$$f^{[1]}(x,y) := \begin{cases} \frac{f(x)-f(y)}{x-y}, & \text{for } x \neq y, \\ f'(x), & \text{for } x = y. \end{cases}$$

Then, according to [115, Theorem 5.1],

$$L_f(A,E) = f^{[1]}\{A, A^H\}(E).$$

Given A and E, a conceptually simple way to compute the Fréchet derivative is via the relation

$$f\left(\begin{bmatrix} A & E \\ 0 & A \end{bmatrix}\right) = \begin{bmatrix} f(A) & L_f(A,E) \\ 0 & f(A) \end{bmatrix}; \tag{3.2}$$

see [127, Theorem 2.1]. As this formula requires evaluating a function of a $2n \times 2n$ matrix (which will typically result in a dense matrix), it is only feasible for small-scale matrices. For the large-and-sparse case, it was long unclear how to efficiently

[29] The function $f^{[1]}$ is a first-order divided difference of f; see [53].

approximate $L_f(A,E)$ at all, and it turns out that this is mostly possible when the direction term E is of low rank. The reason is that then (under suitable assumptions on f), it is also possible to accurately approximate $L_f(A,E)$ by a low-rank matrix; see Section 3.1.1 below, which summarizes the approach developed in [S11].[30] In addition to addressing this computational question, we also discuss several other contributions to theory and applications of Fréchet derivatives: Section 3.1.2 discusses an application of the Fréchet derivative in the analysis of complex networks [S15], where it can be used to rank edges according to their importance in a network or determine suitable up-/downdates which increase/decrease the network's communicability. Sections 3.1.3 and 3.1.4 deal with *higher-order Fréchet derivatives*, which have their main application in defining *level-2 condition numbers*.[31]. The works [S1] and [S14] on which these sections build develop new theoretical results (in the form of explicit formulas or bounds for—possibly structured—level-2 condition numbers) as well as a computational method for approximating higher-order Fréchet derivatives, which improves over all state-of-the-art methods available before. Finally, Section 3.1.5 discusses the generalization of the matrix function Fréchet derivative to certain functions of third-order tensors, so called *t-functions* [123].

3.1.1 P. Kandolf, A. Koskela, S. D. Relton, and M. Schweitzer, *Computing low-rank approximations of the Fréchet derivative of a matrix function using Krylov subspace methods*, Numer. Linear Algebra Appl., 28 (2021)

Ourpaper [S11] introduces a Krylov subspace method for approximating $L_f(A,E)$ when $E = \boldsymbol{b}\boldsymbol{c}^H$ is of rank one (or more generally of low rank). The basis for this method is the integral representation (3.1), which in the case of the Fréchet derivative reduces to the simpler form

$$L_f(A,\boldsymbol{b}\boldsymbol{c}^H) = \frac{1}{2\pi i}\int_\Gamma f(z)(zI-A)^{-1}\boldsymbol{b}\boldsymbol{c}^H(zI-A)^{-1}\,\mathrm{d}z, \tag{3.3}$$

where Γ is a contour that winds around $\operatorname{spec}(A)$ in counterclockwise direction.

Interpreting (3.3) as the integral over solutions of shifted linear systems with A and A^H, a computational method for approximating $L_f(A,\boldsymbol{b}\boldsymbol{c}^H)$ arises by replacing these solutions by their respective Krylov approximations. To be precise,

[30] We note that this is very similar to the case of large-scale Sylvester or Lyapunov matrix equations, which can only be solved by computational methods if the right-hand side term—and thus also the solution—is of low rank.

[31] A "level-2 condition number" is the condition number of the condition number and thus measures how sensitive the condition number itself is to perturbations in the data.

assume that we have Arnoldi decompositions

$$\begin{aligned} AU_m &= U_m G_m + g_{m+1,m} \boldsymbol{u}_{m+1} \boldsymbol{e}_m^H, \\ A^H V_m &= V_m H_m + h_{m+1,m} \boldsymbol{v}_{m+1} \boldsymbol{e}_m^H, \end{aligned}$$

available, where U_m and V_m are orthonormal bases of the Krylov subspaces $\mathcal{K}_m(A, \boldsymbol{b})$ and $\mathcal{K}_m(A^H, \boldsymbol{c})$, respectively. Then the standard FOM approximations $\boldsymbol{x}_m(z)$ for $(zI - A)\boldsymbol{x}(z) = \boldsymbol{b}$ and $\boldsymbol{y}_m(z)$ for $(zI - A^H)\boldsymbol{y}(z) = \boldsymbol{c}$ are given by

$$\boldsymbol{x}_m(z) = \|\boldsymbol{b}\| U_m (zI - G_m)^{-1} \boldsymbol{e}_1, \tag{3.4}$$

$$\boldsymbol{y}_m(z) = \|\boldsymbol{c}\| V_m (zI - H_m)^{-1} \boldsymbol{e}_1. \tag{3.5}$$

Inserting (3.4) and (3.5) into (3.3) yields an approximation $L_m \approx L_f(A, \boldsymbol{b}\boldsymbol{c}^H)$,

$$L_m := \frac{\|\boldsymbol{b}\|\|\boldsymbol{c}\|}{2\pi i} \int_\Gamma f(z) U_m (zI - G_m)^{-1} \boldsymbol{e}_1 \boldsymbol{e}_1^H (zI - H_m)^{-H} V_m^H \,\mathrm{d}t =: U_m X_m V_m^H.$$

According to [S11, Lemma 1], the *core factor* X_m can be computed via the relation

$$f\left(\begin{bmatrix} G_m & \|\boldsymbol{b}\|\|\boldsymbol{c}\| \boldsymbol{e}_1 \boldsymbol{e}_1^H \\ 0 & H_m^H \end{bmatrix} \right) = \begin{bmatrix} f(G_m) & X_m \\ 0 & f(H_m^H) \end{bmatrix}, \tag{3.6}$$

which is very reminiscent of the basic "block formula" (3.2) for the Fréchet derivative. By employing this relation, it is not necessary to numerically evaluate the integral representation above. Collecting the above steps gives rise to a basic Krylov subspace method for approximating $L_f(A, \boldsymbol{b}\boldsymbol{c}^H)$ at the cost of $2m$ matrix vector products, $2m$ Gram–Schmidt orthogonalization steps and the evaluation of the function of a $2m \times 2m$ block upper triangular matrix. Note that there are many applications, in which it is not necessary to explicitly form L_m (which becomes infeasible for large n), so that it can be kept in factored form, storing only U_m, X_m and V_m. For example, matrix vector products with the Fréchet derivative can be performed very efficiently using this representation; see also Section 3.1.2 below for more sophisticated algorithms exploiting the factorized representation of L_m. We note that a Krylov algorithm for the approximation of bivariate matrix functions which is mathematically equivalent to our basic method from [S11] was independently proposed in [116].

In addition, [S11] discusses several algorithmic enhancements of the basic method outlined above, e.g., simplifications for the case of Hermitian A as well as the use of block methods or rational Krylov approaches [S11, Sections 2.1–2.5] as well as accurate a posteriori error estimates [S11, Section 5]. On the theoretical side, we prove convergence of the method for a wide range of matrix and function classes, by essentially combining techniques used in the convergence analysis of Krylov methods for matrix functions [80] with those used for Krylov methods for matrix

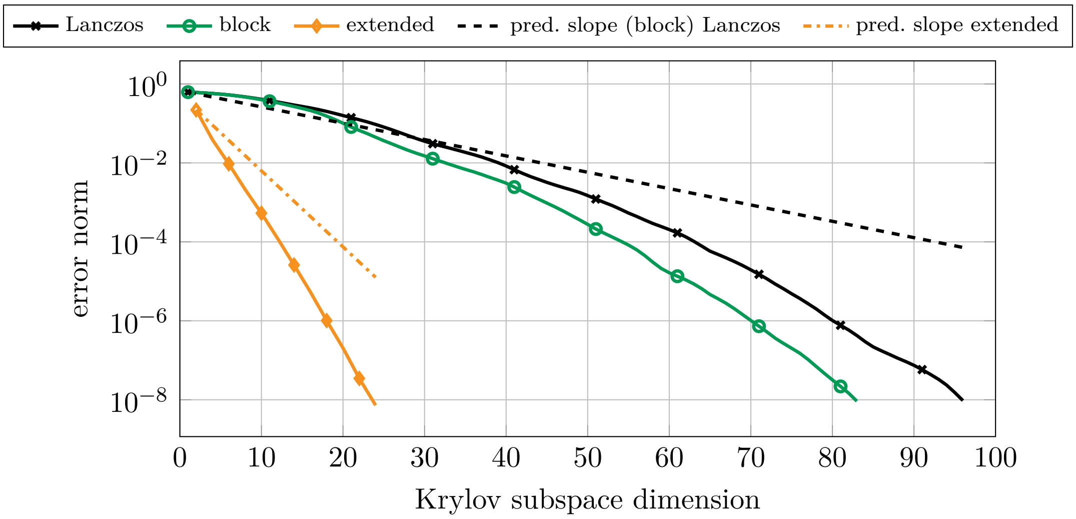


Figure 3.1: Error norm and (a priori) error bounds when approximating $L_f(A,E)$ by several of the methods proposed in [S11]. Here, A is the discretization of the two-dimensional Laplace operator, $f(z)=z^{-1/2}$ and $E=\boldsymbol{b}\boldsymbol{c}^H$ is a random rank-one matrix. *Originally [S11, Figure 5].*

equations [114]. We recall here the result of [S11, Theorem 4], which states that for Hermitian positive definite A and f a Cauchy–Stieltjes function,

$$\|L_f(A,\boldsymbol{b}\boldsymbol{c}^H)-L_m\| \leq 4\|\boldsymbol{b}\|\|\boldsymbol{c}\|\;|f'(\lambda_{\min})|\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^m, \tag{3.7}$$

where $\lambda_{\min}$ is the smallest eigenvalue of A and κ denotes the Euclidean norm condition number of A.

Numerical experiments—with matrices taken both from benchmark collections and from an application involving the simulation of nuclear transmutation—confirm the effectiveness and numerical accuracy of the developed approach. As an example, we report in Figure 3.1 the results of a simple experiment, in which we approximate the Fréchet derivative of the inverse square root at the discretized two-dimensional Laplace operator, with a randomly drawn rank-one direction $E=\boldsymbol{b}\boldsymbol{c}^H$. We test the basic Lanczos method [S11, Section 2.1], a block Lanczos method which builds a single block Krylov space with starting (block) vector $[\boldsymbol{b},\boldsymbol{c}]\in\mathbb{R}^{n\times 2}$ instead of two separate Krylov spaces [S11, Section 2.4] and an extended Krylov method [S11, Section 2.5]. We report both the actual error as well as the slope of the a priori convergence bounds from [S11, Theorems 4 & 6]. Both the polynomial and the extended Krylov method converge faster than predicted by the a priori bound, a typical phenomenon also known for similar bounds for Krylov approximations of $f(A)\boldsymbol{b}$. In particular, an a priori bound like (3.7) cannot predict the superlinear convergence that occurs in this example.[32] As

[32] The same is true, e.g., for the textbook convergence bound of the conjugate gradient method, in the context of solving linear systems, which involves the same convergence factor.

expected, the extended Krylov subspace method converges in much fewer iterations than the polynomial methods (but it requires the solution of linear systems instead of only matrix vector products). The polynomial block Krylov method works a bit more efficiently than the non-block method, at essentially the same computational cost per iteration.[33]

3.1.2 M. Schweitzer, *Sensitivity of matrix function based network communicability measures: Computational methods and a priori bounds*, SIAM J. Matrix Anal. Appl., 44 (2023) – Sections 2–4

In [S15], we discuss an application of the Fréchet derivative in network science. Assume that A_G is the adjacency matrix of a graph $G = (\mathcal{V}, \mathcal{E})$, representing a complex network (e.g., a social network, traffic network, ...). As outlined in Example 1.4, total communicability $\mathrm{TC}(G) = \boldsymbol{1}^H \exp(A_G)\boldsymbol{1}$ (cf. (1.5)) is a common measure for how well the network is able to transport information (or goods, depending on the application).

An interesting question in many areas of network science (e.g., when trying to design optimized networks or in vulnerability analysis) is how communicability measures such as (1.5) react to changes in the network, with the most common change being the addition or removal of edges.

One way to approach this question is to study *total network sensitivity* [56], defined as

$$\mathrm{TS}_{ij}(G) := \boldsymbol{1}^H L_{\exp}(A_G, E_{ij})\boldsymbol{1}, \tag{3.8}$$

where $E_{ij} = \boldsymbol{e}_i \boldsymbol{e}_j^H$. Clearly, from the multivariate chain rule, $\mathrm{TS}_{ij}(G)$ measures the rate of change of total communicability when the weight of edge (i, j) is modified.[34]

One fundamental limitation of the practical applicability of this sensitivity concept introduced in [56] is that the direction term in (3.8) depends on the considered edge, so that computing all sensitivity values requires evaluating a quadratic form involving a Fréchet derivative *for each individual edge*: In most applications, G will be a large, sparse graph with n nodes and $\mathcal{O}(n)$ edges. Thus, when one tries

[33] In fact, the product between A and a block vector of size $n \times 2$ instead of two matrix vector products might actually be cheaper due to more advantageous memory access. The orthogonalization in a block method is typically a bit more expensive than in a method operating just on vectors, though.

[34] Note that pretending that edge weights can be changed freely—although G might originally be an unweighted graph—is common practice as it leads to continuous optimization problems which are easier to tackle than the discrete optimization problems that would otherwise arise.

to identify an "optimal" new edge to add to the graph, there are $\mathcal{O}(n^2)$ candidate edges. Even assuming the ideal setting that one evaluation of the Fréchet derivative is possible with cost $\mathcal{O}(n)$—e.g., by $\mathcal{O}(1)$ steps of the Krylov method from [116, S11]—this means that finding the edge with highest total sensitivity value will have a cost of $\mathcal{O}(n^3)$, which might already be infeasible even for medium-scale networks.

The crucial observation for overcoming the limitation mentioned in the previous paragraph is that according to [S15, Theorem 2.3 & Corollary 2.4], an alternative way of defining and computing total network sensitivity is

$$\mathrm{TS}_{ij}(G) = [L_{\exp}(A_G^H, \boldsymbol{1}\boldsymbol{1}^H)]_{ij}. \tag{3.9}$$

The important feature of (3.9) is that the direction term in the Fréchet derivative is independent of the edge (i, j) under consideration (and is still of rank one). Thus, a single call of the Krylov algorithm from [116, S11] suffices to approximate all sensitivities, instead of $\mathcal{O}(n^2)$ calls. One additional difficulty that one needs to overcome to be able to deal with large-scale networks, though, is that explicitly forming the approximation $L_m \approx L_{\exp}(A_G^H, \boldsymbol{1}\boldsymbol{1}^H)$ is typically infeasible as it requires $\mathcal{O}(n^2)$ storage. Thus, if one is interested in finding the best edges to add to the network (or the most important existing edges), one needs to find the largest entries of L_m without explicitly forming it. To do so, the algorithm introduced in [S15] leverages a maximum element estimator for implicitly given matrices from [96] which is based on a subgradient method and only requires performing matrix-vector products.

An additional difficulty arises from the fact that one is typically not interested in simply finding the largest elements in L_m, but rather wants to restrict to certain "candidate edges".[35] Therefore, one cannot work with L_m directly but must instead use a masked version, $L_m^{\text{masked}} := M \odot L_m$, where the binary mask M marks candidate edges and $\odot$ denotes the Hadamard (or element-wise) matrix product. To apply the maximum element estimator from [96] for up- or downdating networks, one must therefore be able to perform an efficient matrix vector product with L_m^{masked}. This is possible based on the well-known result that

$$(A \odot BC^H)\boldsymbol{x} = \sum_{i=1}^{r} D_{\boldsymbol{b}_i} A \overline{D}_{\boldsymbol{c}_i} \boldsymbol{x}, \tag{3.10}$$

where $\boldsymbol{b}_i, \boldsymbol{c}_i, i = 1, \ldots, r$ are the columns of B and C, respectively, and $D_{\boldsymbol{y}}$ is the diagonal matrix with the entries of the vector $\boldsymbol{y}$ on the diagonal. Thus, as long as the matrix A in (3.10) allows an efficient matrix vector product and the rank of BC^H is not too large, efficient matrix vector products with $A \odot BC^H$

[35] The most common cases of candidate sets—corresponding to up- and downdates, respectively—being either all existing or all non-existing (or virtual) edges.

	n	200	400	800	1600	3200	6400	12800
	avg. deg.	9.88	10.0	10.4	10.9	11.0	11.2	11.2
Alg. from [S15]	Kryl. it.	14	17	18	22	22	23	25
	HR it.	4	2	4	2	3	5	2
	time	0.02s	0.03s	0.06s	0.08s	0.26s	0.70s	0.73s
Alg. from [56]	Kryl. it.	12.7	14.8	15.7	14.9	*	*	*
	time	40s	216s	1049s	5417s	*	*	*

Table 3.1: Results obtained for random geometric graphs of varying size. “Kryl. it.” refers to number of iterations in the Krylov algorithm from [116,S11]. Note that the algorithm from [56] requires many calls of this method, and we report the average number of iterations across all runs. ”HR it.” refers to number of iterations in [96, Algorithm 5.2]. Entries marked with * indicate that the method did not terminate within two hours. *Originally [S15, Table 3.3].*

are possible, keeping BC^H in factored form. Fortunately, this is exactly the case in the situation at hand. Typically, $m = \mathcal{O}(1)$ Krylov steps are sufficient for approximating $L_{\exp}(A_G^H, \boldsymbol{1}\boldsymbol{1}^H)$ accurately enough, so that L_m is of low rank. The binary mask M—which takes the role of A in (3.10)—is either the binary adjacency matrix A_G (for existing edges) or $\boldsymbol{1}\boldsymbol{1}^H - (A_G + I)$ (for virtual edges), both of which allow an efficient matrix vector product when G is a sparse graph.

Combining all the elements outlined above, [S15] proposes an algorithm which (under mild assumptions) can be expected to scale linearly with the size n of the graph, thus making it feasible to use also for large scale networks. Numerical experiments—both on artificially generated graphs and on real-world complex networks from various applications—confirm almost perfect linear scaling of the method.

As an example, we report in Table 3.1 the results of an experiment where we estimate the $p = 10$ virtual edges with maximum sensitivity in random geometric graphs of varying sizes in order to investigate the scaling behavior of the method and compare it to the baseline method from [56]. The random geometric graphs are constructed such that the average degree of their nodes is roughly 10; Figure 3.2 shows an example of such a graph for $n = 400$. The results clearly indicate the superiority over the baseline method and confirm the linear scaling; see [S15, Example 3.5] for details on parameter choices and algorithmic setup.

In addition to the content summarized above, [S15] also contains a discussion of how to extend the sensitivity concept from total network communicability to other centrality and communicability measures like *subgraph centrality* [75] and the Estrada index [71,73] (cf. Example 1.4) and to modifications of nodes (instead of edges).

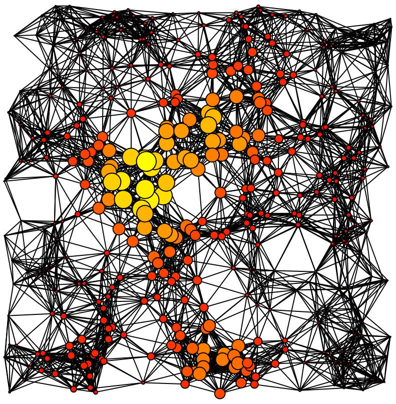


Figure 3.2: Illustration of a random geometric graph with $n = 400$ nodes. The size and color of nodes encodes their total communicability (with lighter colors corresponding to larger values). *Originally [S15, Figure 3.2].*

The second part of [S15] discusses decay in the entries of Fréchet derivatives, which allows to obtain a priori bounds on the sensitivities. As this part of the paper thematically better fits into the content of Chapter 4, we discuss it in more detail in Section 4.1.4.

3.1.3 M. Schweitzer, *Integral representations for higher-order Fréchet derivatives of matrix functions: Quadrature algorithms and new results on the level-2 condition number*, Linear Algebra Appl., 656 (2023)

Higher-order Fréchet derivatives of matrix functions can be defined in a recursive fashion. Given a matrix A, a sufficiently smooth function f and letting $L_f^{(1)}(A,E) := L_f(A,E)$, the *kth Fréchet derivative of f at A* is defined as the unique multilinear function $L_f^{(k)}(A,\cdot,\ldots,\cdot)$ of the matrices $E_i, i = 1,\ldots,k$ that satisfies

$$\begin{aligned} L_f^{(k-1)}(A+E_k,E_1,\ldots,E_{k-1}) - L_f^{(k-1)}(A,E_1,\ldots,E_{k-1}) \\ = \quad L_f^{(k)}(A,E_1,\ldots,E_k) + o(\|E_k\|). \end{aligned}$$

Applications of higher-order Fréchet derivatives include the computation of the level-2 condition number [95] (also see below) as well as solving nonlinear equations in Banach spaces; see, e.g., [5]. We also refer to [141] for recent work on a unifying, more general concept for higher-order derivatives of matrix functions, with applications in quantum perturbation theory.

A level-2 condition number measures how sensitive the condition number itself is to perturbations. For a matrix function, the *(absolute) level-two condition number* can straightforwardly be defined via

$$\operatorname{cond}^{(2)}(f, A) := \lim_{\varepsilon \to 0} \sup_{\|Z\| \leq \varepsilon} \frac{|\operatorname{cond}(f, A+Z) - \operatorname{cond}(f, A)|}{\varepsilon},$$

a form which is not well-suited for actual numerical computations. In [95], a bound for the level-2 condition number (with respect to the Frobenius norm) is derived, involving the second-order Fréchet derivative. To be specific,

$$\operatorname{cond}^{(2)}(f, A) \leq \max_{\|E_2\|_F = 1} \max_{\|E_1\|_F = 1} \|L_f^{(2)}(A, E_1, E_2)\|_F = \|K_f^{(2)}(A)\|,$$

where $K_f^{(2)}(A)$ is the *Kronecker matrix*, a representation of the second-order Fréchet derivative as an $n^4 \times n^2$ matrix; see [95, Section 4]. In general, it is not possible to compute the exact value of the level-2 condition number (neither analytically nor numerically) except for certain special cases, which are discussed in [95, Section 5].

In [S14], we derive a new integral representation for the higher-order Fréchet derivative. This representation is then used to both extend the class of functions for which it is possible to exactly compute the level-2 condition number and to derive an efficient algorithm based on numerical quadrature for approximating the higher-order Fréchet derivative.

The main result [S14, Theorem 2]—which is based on an explicit representation of the higher-order Fréchet derivative of the resolvent—is the following. If f is analytic on and inside a contour Γ that winds around $\operatorname{spec}(A)$ exactly once,

$$L_f^{(k)}(A, E_1, \ldots E_k) = \frac{1}{2\pi i} \int_\Gamma \sum_{\pi \in S_k} f(\zeta) M_\pi(\zeta; A, E_1, \ldots, E_k) \, \mathrm{d}\zeta, \tag{3.11}$$

where S_k denotes the *symmetric group of degree* k, i.e., the set of all permutations of $\{1, \ldots, k\}$ and

$$\begin{aligned} &M_\pi(\zeta; A, E_1, \ldots, E_k) \\ &\quad = (\zeta I - A)^{-1} E_{\pi(1)} (\zeta I - A)^{-1} E_{\pi(2)} (\zeta I - A)^{-1} \cdots E_{\pi(k)} (\zeta I - A)^{-1}. \end{aligned}$$

Using (3.11) allows to derive a new upper bound for the level-2 condition number whenever A is Hermitian positive definite and f is a Cauchy–Stieltjes function or $f(z) = zg(z)$ with g a Cauchy–Stieltjes function. If further the smallest eigenvalue $\lambda_{\min}$ of A is simple, this upper bound exactly agrees with a lower bound proven in [95, Theorem 5.5]. Thus, in that case, the new upper bound actually gives an explicit formula for the level-2 condition number. We now summarize this result.

Theorem 3.2. (Theorem 7 and Corollary 8 in [S14]) *Let f be a Cauchy–Stieltjes function or a function of the form $f(z) = zg(z)$, where g is a Cauchy–Stieltjes function and let A be Hermitian positive definite with smallest eigenvalue $\lambda_{\min}$. Then*

$$\mathrm{cond}^{[2]}(f, A) \leq |f''(\lambda_{\min})|.$$

If further $\lambda_{\min}$ is a simple eigenvalue of A, then

$$\mathrm{cond}^{[2]}(f, A) = |f''(\lambda_{\min})|.$$

For actually computing the higher-order Fréchet derivative, all commonly used methods are based on the observation that (under suitable assumptions on the smoothness of f), $L_f^{(k)}(A, E_1, \dots, E_k)$ is equal to the upper right $n \times n$ block of $f(X_k)$, where

$$X_k = \begin{cases} A & k = 0 \\ I_2 \otimes X_{k-1} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \otimes I_{2^{k-1}} \otimes E_k & k \geq 1, \end{cases}$$

and $\otimes$ denotes the Kronecker product.[36] The (block upper triangular) matrix X_k is of size $2^k n \times 2^k n$, so that working with it (and even storing it) quickly becomes prohibitively costly, already for moderate values of k. Existing algorithms either directly compute $f(X_k)$ [95, Algorithm 3.6], or use a complex step approximation [124] to reduce the computation for X_k to one that only involves X_{k-1} or X_{k-2} [3, 167]. As an alternative, discretizing (3.11) by a suitable numerical quadrature rule (which will typically depend on f) allows to approximate $L_f^{(k)}(A, E_1, \dots, E_k)$ working only with matrices of size $n \times n$. However, the number of terms in the sum in (3.11) grows as $k!$, so that this approach is also infeasible if k gets too large. By using a careful implementation and exploiting the fact that many quantities can be reused across several computations, this approach often still offers significant benefits over established methods.

The numerical experiments in [S14] show that the quadrature-based approach typically runs significantly faster than the alternatives for values of k between 2 and 6 when the direction terms E_i are unstructured or between 2 and 8 when the direction terms are of low rank. In the latter case, further optimizations are possible; see [S14, Section 4.3]. We exemplarily report the results of [S14, Experiment 15] in Figure 3.3, where we compare the performance of our quadrature based method with the baseline method from [95] and the recently proposed complex step approximation from [3]. Here, the direction terms are outer products

[36] This formula generalizes the "2×2 block formula" (3.2) for the first-order Fréchet derivative.

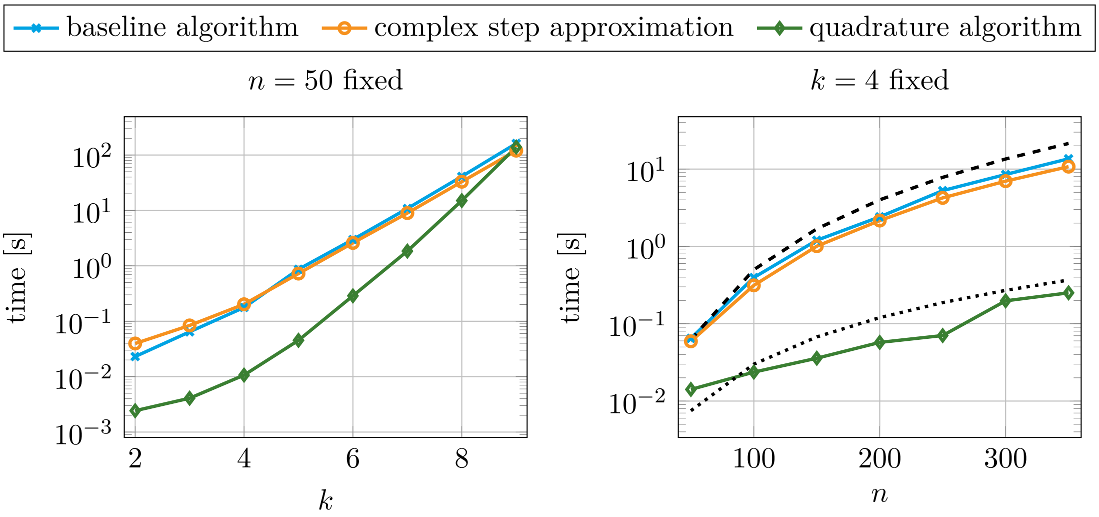


Figure 3.3: Run-time comparison of the different algorithms for $n = 50$ and varying k (left) and for $k = 4$ and varying n (right). In both cases, the matrix $A \in \mathbb{R}^{n\times n}$ is the tridiagonal matrix generated by the command `A = gallery('lesp',n)` and the matrices $E_1, \dots, E_k$ are outer products of canonical unit vectors. The dotted and dashed line in the right plot indicate quadratic and cubic scaling with respect to n, respectively. *Originally [S14, Figure 2].*

of canonical unit vectors.[37] One can clearly observe that our quadrature algorithm shows a better scaling behavior when n is increased (quadratic instead of cubic), and that for values of k ranging from 2 to 6, it is about one order of magnitude faster than the competing methods (and still faster for $k = 7, 8$). We refer to [S14, Experiment 15] for further details regarding experimental setup, parameter choices etc.

To conclude this section, we note that an additional benefit of our quadrature algorithm—which we haven't explored further—is that it is trivially parallelizable, as the computations for individual quadrature nodes are completely independent, so that even larger gains can be expected in a parallel computing environment.

3.1.4 B. Arslan, S. D. Relton, and M. Schweitzer, *Structured level-2 condition numbers of matrix functions*, Electron. J. Linear Algebra, 40 (2024)

Our paper [S1] also deals with level-2 condition numbers of matrix functions, which were already discussed in Section 3.1.3 above. In contrast to the usual level-2 condition number considered in that section, this work focuses on so-called

[37] This is a typical case which, e.g., occurs in algorithms for the higher-order condition number.

M	**Automorph. Gr.** ($\mathbb{G}$)	**Jordan Algebra** ($\mathbb{J}$)	**Lie Algebra** ($\mathbb{L}$)
I_n	Orthogonal	Symmetric	Skew-symmetric
$\Sigma_{p,q}$	Pseudo-orthogonal	Pseudo-symmetric	Pseudo skew-symmetric
R_n	Perplectic	Persymmetric	Perskew-symmetric
J_n	Symplectic	Skew-Hamiltonian	Hamiltonian

Table 3.2: Choices for M from (3.12) leading to well-known matrix classes $\mathbb{J}$, $\mathbb{L}$ and $\mathbb{G}$. *Adapted from [S1, Table 1].*

structured condition numbers, i.e., instead of allowing arbitrary perturbations of the input, one determines the sensitivity only with respect to perturbations which keep a certain structure in the matrix intact. This allows to better analyze *structure-preserving* algorithms for the computation of $f(A)$ or its condition number, which are known to often yield much more accurate results than their general purpose counterparts.

In [S1], we mostly consider matrices from Lie algebras $\mathbb{L}$, Jordan algebras $\mathbb{J}$ and automorphism groups $\mathbb{G}$ corresponding to a scalar product $\langle \cdot, \cdot \rangle_M$, defined via

$$\langle x, y\rangle_M = \begin{cases} x^T M y, & \text{for real or complex bilinear forms,} \\ x^H M y, & \text{for sesquilinear forms.} \end{cases}$$

The matrix structures of interest then arise via

$$\begin{aligned} \mathbb{J} &:= \{A \in \mathbb{K}^{n\times n} \mid A^\star = A\}, \\ \mathbb{L} &:= \{A \in \mathbb{K}^{n\times n} \mid A^\star = -A\}, \\ \mathbb{G} &:= \{A \in \mathbb{K}^{n\times n} \mid A^\star = A^{-1}\}, \end{aligned}$$

respectively, where $A^\star$ denotes the adjoint of A with respect to $\langle \cdot, \cdot \rangle_M$ and $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$. Common choices of M that give rise to practically important matrix structures are

$$\Sigma_{p,q} = \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix}, \quad R_n = \begin{bmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{bmatrix}, \quad \text{and} \quad J_n = \begin{bmatrix} 0 & I_{n/2} \\ -I_{n/2} & 0 \end{bmatrix}, \tag{3.12}$$

where for $\Sigma_{p,q}$ we have $p+q=n$; see Table 3.2. Additionally, we consider the class of quasi-triangular matrices, as it can also be handled with similar techniques.

The structured level-2 condition number of a matrix function can be defined as

$$\mathrm{cond}^{[2]}_{\mathrm{struc}}(f, A) = \lim_{\varepsilon \to 0} \sup_{\substack{A+Z \in \mathbb{S}_{\mathcal{M}} \\ \|Z\| \le \varepsilon}} \frac{|\mathrm{cond}_{\mathrm{struc}}(f, A+Z) - \mathrm{cond}_{\mathrm{struc}}(f, A)|}{\varepsilon}.$$

where now $\mathbb{S}_{\mathcal{M}}$ denotes any smooth matrix manifold (which will in practice typically correspond to one of the choices outlined above) and $\mathrm{cond}_{\mathrm{struc}}$ is the usual structured (level-1) condition number. The main result of [S1] is a novel upper bound for the structured level-2 condition number,

$$\mathrm{cond}^{[2]}_{\mathrm{struc}}(f, A) \leq \|(B_{\mathcal{M}} B_{\mathcal{M}}^{+} \otimes I_{n^2}) K_f^{(2)}(A) B_{\mathcal{M}} B_{\mathcal{M}}^{+}\|_2, \qquad (3.13)$$

where the columns of $B_{\mathcal{M}}$ span $T_A\mathbb{S}_{\mathcal{M}}$, the tangent space[38] of $\mathbb{S}_{\mathcal{M}}$ at A; see [S1, Lemma 3.1].

Based on construction formulas[39] for bases of tangent spaces, an algorithm for computing an upper bound of the structured level-2 condition number is presented in [S1, Algorithm 1]. In numerical experiments, upper bounds for the structured and unstructured level-2 condition number are compared to each other. As only comparing upper bounds does not necessarily allow to draw conclusions about the actual quantities (as long as there is no precise knowledge on the tightness of the bounds), we also obtain *lower* bounds for the respective condition numbers by employing methods from continuous optimization. The numerical results both confirm that in most cases the upper bound (3.13) is quite tight (as it is close to the lower bound) and that the structured level-2 condition number is often many orders of magnitude smaller than the unstructured one (in particular for ill-conditioned test matrices).

As an example, Figure 3.4 shows results obtained for the matrix logarithm of orthogonal and symplectic test matrices which are constructed using Jagger's toolbox [109]. For the orthogonal test set, the upper bound for the structured condition number lies on average about one order of magnitude below the unstructured one, and the respective lower bounds confirm that the unstructured condition number is indeed guaranteed to be smaller than the structured one. For the symplectic test set, the difference is a lot more pronounced. While the unstructured condition number grows proportionally to the two-norm condition number, the structured condition number roughly stays constant across all matrices. The lower bounds are again quite close to the upper bounds, indicating that those are rather tight.[40]

In addition to these results [S1, Section 3.3] presents explicit formulas for the structured level-2 condition number of the matrix exponential of Hermitian or skew-Hermitian matrices.

[38] The tangent space at A is defined as $T_A\mathbb{S}_{\mathcal{M}} := \{E \in \mathbb{K}^{n\times n} \mid \exists \text{ a smooth curve } \gamma : \mathbb{K} \to \mathbb{S}_{\mathcal{M}} \text{ with } \gamma(0) = A, \gamma'(0) = E\}$.

[39] Based on [7] for Jordan and Lie algebras and automorphism groups and on [2] for the quasi-triangular case.

[40] For the most ill-conditioned test problem, the computed lower bound lies *above* the upper bound, which should of course be impossible. This is caused by accumulation of errors in the optimization algorithm due to the very bad conditioning of the matrix.

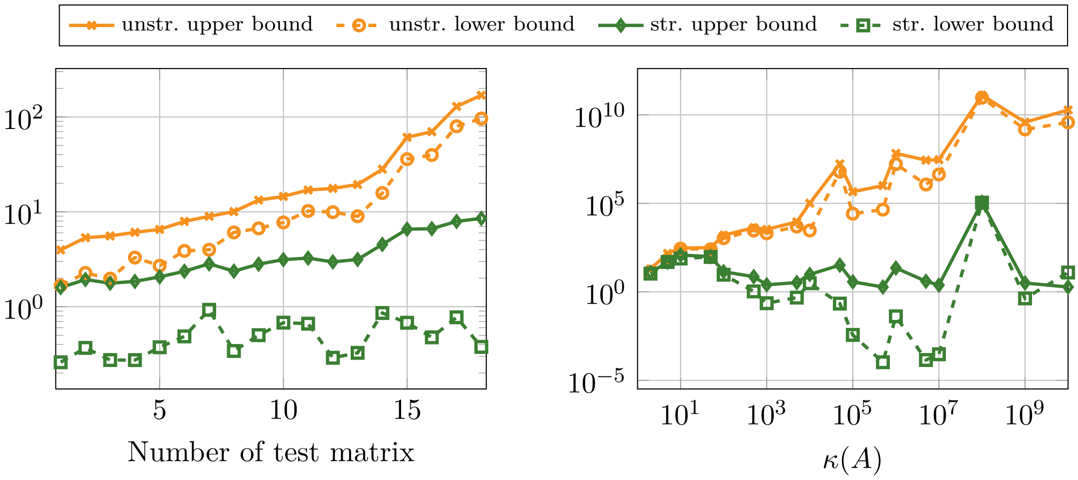


Figure 3.4: Structured and unstructured level-2 condition numbers for the matrix logarithm, comparing upper and lower bounds. *Left:* Orthogonal matrices. *Right:* Symplectic matrices. *Adapted from [S1, Figure 1].*

3.1.5 K. Lund and M. Schweitzer, *The Fréchet derivative of the tensor t-function*, Calcolo, 60 (2023)

Recently, [123] has proposed a generalization of matrix functions to functions of third-order tensors (see Figure 3.5), based on the t-product framework [44, 111, 112]. This framework defines a way to multiply third-order tensors, and is based on viewing them as stacks of frontal slices (as in Figure 3.5(d)): Given two tensors $\mathcal{A} \in \mathbb{C}^{n\times m\times p}, \mathcal{B} \in \mathbb{C}^{m\times s\times p}$, the t-product is defined as

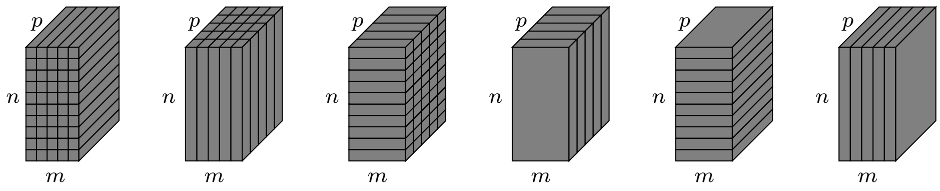
$$\mathcal{A} * \mathcal{B} := \texttt{fold}(\texttt{bcirc}(\mathcal{A})\texttt{unfold}(\mathcal{B})),$$

where the operations `unfold` and `fold` transform the tensor $\mathcal{B}$ into a block vector of size $mp \times s$ and vice versa, i.e.,

$$\texttt{unfold}(\mathcal{B}) := \begin{bmatrix} B^{(1)} \\ B^{(2)} \\ \vdots \\ B^{(p)} \end{bmatrix}, \text{ and } \texttt{fold}(\texttt{unfold}(\mathcal{B})) := \mathcal{B},$$

`bcirc` turns $\mathcal{A}$ into a block-circulant matrix of size $np \times mp$,

$$\texttt{bcirc}(\mathcal{A}) := \begin{bmatrix} A^{(1)} & A^{(p)} & A^{(p-1)} & \cdots & A^{(2)} \\ A^{(2)} & A^{(1)} & A^{(p)} & \cdots & A^{(3)} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ A^{(p-1)} & \ddots & \ddots & A^{(1)} & A^{(p)} \\ A^{(p)} & A^{(p-1)} & \cdots & A^{(2)} & A^{(1)} \end{bmatrix}$$



(a) (b) (c) (d) (e) (f)

Figure 3.5: Different views of a third-order tensor $\mathcal{A} \in \mathbb{C}^{n \times m \times p}$. (a) tube fibers: $\mathcal{A}(:, j, k)$, (b) column fibers: $\mathcal{A}(i, :, k)$, (c) row fibers: $\mathcal{A}(i, j, :)$, (d) frontal slices: $\mathcal{A}(i, :, :)$, (e) lateral slices: $\mathcal{A}(:, j, :)$, (f) horizontal slices: $\mathcal{A}(:, :, k)$. *Originally [S12, Figure 1].*

and $A^{(k)}$ and $B^{(k)}$, $k = 1, \dots, p$ denote the frontal slices of $\mathcal{A}$ and $\mathcal{B}$, respectively.[41] Clearly, the t-product is associative.

Using these definitions, the action of the *tensor t-function* f of $\mathcal{A} \in \mathbb{C}^{n \times n \times p}$ on another tensor $\mathcal{B} \in \mathbb{C}^{n \times s \times p}$ is defined as

$$f(\mathcal{A}) * \mathcal{B} := \texttt{fold}(f(\texttt{bcirc}(\mathcal{A})) \cdot \texttt{unfold}(\mathcal{B})). \tag{3.14}$$

The t-function $f(\mathcal{A})$ itself is obtained by taking $\mathcal{B}$ to be the identity tensor,

$$\begin{aligned} f(\mathcal{A}) &:= \texttt{fold}(f(\texttt{bcirc}(\mathcal{A})) \cdot \texttt{unfold}(\mathcal{I}_{n \times n \times p})) \\ &= \texttt{fold}\big(f(\texttt{bcirc}(\mathcal{A}))\boldsymbol{E}_1^{np \times n}\big), \end{aligned}$$

where $\boldsymbol{E}_1^{np \times n} = \boldsymbol{e}_1 \otimes I_n$ with $\boldsymbol{e}_1 \in \mathbb{C}^p$. This so-called *t-function* inherits many of the usual properties of matrix functions; see [123, Section 2.2]. In [S12], we investigate the Fréchet derivative of the t-function, which can naturally be represented as

$$L_f(\mathcal{A}, \mathcal{C}) = \texttt{fold}\big(L_f(\texttt{bcirc}(\mathcal{A}), \texttt{bcirc}(\mathcal{C}))\boldsymbol{E}_1^{np \times n}\big);$$

see [S12, Lemma 2]. Besides studying elementary properties of this *t-Fréchet derivative*, we also derive several useful representations for it. For example, when f is analytic, the Fréchet derivative of the t-function can be written as

$$L_f(\mathcal{A}, \mathcal{C}) = \frac{1}{2\pi i} \int_\Gamma f(\zeta)(\zeta\mathcal{I} - \mathcal{A})^{-1} * \mathcal{C} * (\zeta\mathcal{I} - \mathcal{A})^{-1} \, \mathrm{d}\zeta, \tag{3.15}$$

which generalizes the integral representation (3.3) for the Fréchet derivative of matrix functions; see [S12, Lemma 5]. The inverse in (3.15) is of course to be understood as inversion with respect to the t-product. Alternatively, generalizing the "block representation" (3.2), the t-Fréchet derivative can be written as

$$L_f(\mathcal{A}, \mathcal{C}) = \texttt{fold}\left(\left[f\left(\begin{bmatrix} \texttt{bcirc}(\mathcal{A}) & \texttt{bcirc}(\mathcal{C}) \\ O_{np \times np} & \texttt{bcirc}(\mathcal{A}) \end{bmatrix} \right) \begin{bmatrix} O_{np \times n} \\ I_n \\ O_{n(p-1) \times n} \end{bmatrix} \right]_{1:np,:} \right)$$

[41] I.e., in MATLAB notation, $A^{(k)} = \mathcal{A}(:, :, k)$ and $B^{(k)} = \mathcal{B}(:, :, k)$.

with $O_{k\times\ell}$ a $k\times\ell$ matrix of all zeros. In [S12, Section 3.4], we propose a Kronecker representation of the Fréchet derivative of the t-function and present algorithms for its efficient computation which exploit symmetries and shift invariances.

We then propose two different methods for approximating $L_f(\mathcal{A},\mathcal{C})$. The first one is a (block) Krylov approach which is applicable when the direction term $\mathcal{C}$ is of low rank and is a straightforward generalization of the Krylov method from [116,S11] described in Section 3.1.1, based on the integral representation (3.15).

The second approach exploits the well-known fact that block-circulant matrices are block-diagonalized by the discrete Fourier transform (DFT) [111, 112], which allows to decouple most computations and obtain an embarrassingly parallel method. The precise result is the following, and its proof is based on a Daleckiĭ-Kreĭn-type result for block diagonal matrices [S12, Theorem 2].

Corollary 3.3. (Corollary 2 in [S12]) *Let $\mathcal{A},\mathcal{C}\in\mathbb{C}^{n\times n\times p}$ and let f be $2np-1$ times continuously differentiable on a region containing* $\operatorname{spec}(\texttt{bcirc}(\mathcal{A}))$. *Further, let*

$$(F_p\otimes I_n)\texttt{bcirc}(\mathcal{A})(F_p^H\otimes I_n)=\mathcal{D}^A$$

and

$$(F_p\otimes I_n)\texttt{bcirc}(\mathcal{C})(F_p^H\otimes I_n)=\mathcal{D}^C$$

where $\mathcal{D}^A=\operatorname{blkdiag}(D_1^A,\dots,D_p^A)$, $\mathcal{D}^C=\operatorname{blkdiag}(D_1^C,\dots,D_p^C)$ *and F_p is the $p\times p$ discrete Fourier transform matrix. Then*

$$L_f(\mathcal{A},\mathcal{C})=\texttt{fold}\left((F_p^H\otimes I_n)\begin{bmatrix}\frac{1}{\sqrt{p}}L_1\\ \vdots\\ \frac{1}{\sqrt{p}}L_p\end{bmatrix}\right),$$

where the diagonal blocks $L_i, i=1,\dots,p$ are given by

$$L_i=L_f(D_i^A,D_i^C),\qquad i=1,\dots,p.$$

As applications of the concept, we show that the condition number of the t-function can be related to the norm of its Fréchet derivative, as in the matrix function case. We also show that our formalism can be used to derive a representation of the gradient of the tensor nuclear norm $\|\mathcal{A}\|_\star$, starting from

$$\|\mathcal{A}\|_\star^2=\operatorname{trace}_{(1)}(\sqrt{\mathcal{A}^T*\mathcal{A}}),$$

a relation recently proven in [14, Lemma 6].

We report several numerical experiments which confirm that the proposed algorithms can be used to efficiently approximate the Fréchet derivative for t-functions of small and medium scale tensors; see [S12, Section 6].

3.2 Updating matrix functions subject to low-rank modifications

Taking into account the fact that matrix function computations are often very costly, the following is a rather obvious and relevant question:

"Assume that $f(A)$ is already known from some previous computation, and A is slightly altered, giving $A + E$. Is it possible to approximate $f(A + E)$, starting from $f(A)$, at a cost that is significantly lower than that of recomputing $f(A+E)$ from scratch?"

Throughout, we will assume that A and E are such that f is analytic on a domain that contains $\operatorname{spec}(A)$ and $\operatorname{spec}(A + E)$, so that in particular both $f(A)$ and $f(A+E)$ are defined, and we assume that "slightly" above means that E is of low rank $r \ll n$. One particular example of an application in which updating matrix functions is of relevance is updating network centrality measures (cf. Example 1.4) after modifying a graph by inserting or removing nodes or edges.

Clearly, the problem outlined above fits into the framework of bivariate matrix functions considered in this chapter, writing

$$f(A + E) - f(A) = f^{\text{diff}}\{A, E\}(I),$$

where

$$f^{\text{diff}}(x, y) := f(x + y) - f(x).$$

For this rather simple special case, the integral representation (3.1) simplifies to

$$f(A + E) - f(A) = \frac{1}{2\pi i} \int_\Gamma f(z)(zI - A)^{-1} E (zI - A - E)^{-1} \,\mathrm{d}z. \tag{3.16}$$

Note that this representation can also straightforwardly be derived from the second resolvent identity [97], without needing the formalism of bivariate matrix functions.

Closed formulas for $f(A + E) - f(A)$ are only available in very particular special cases, the by far most well-known one certainly being the Sherman–Morrison–Woodbury [151, 169] formula for rank-one updates of the inverse,

$$(A + \boldsymbol{b}\boldsymbol{c}^H)^{-1} - A^{-1} = -\frac{A^{-1}\boldsymbol{b}\boldsymbol{c}^H A^{-1}}{1 + \boldsymbol{c}^H A^{-1} \boldsymbol{b}}. \tag{3.17}$$

A generalization of the Sherman–Morrison–Woodbury formula to rational functions other than the inverse is given in [31, Theorem 3]. To fix notation, let

$r(z) = p(z)/q(z)$ with polynomials $p(z) = \sum_{i=0}^{m_p} \alpha_i z^i$ and $q(z) = \sum_{i=0}^{m_q} \beta_i z^i$ and set $m = \max\{m_p, m_q\}$. Then, provided that $r(A)$ and $r(A + \boldsymbol{bc}^H)$ are well-defined,

$$r(A + \boldsymbol{bc}^H) - r(A) = XY^H, \tag{3.18}$$

where X, Y are defined by $X = q(A)^{-1}K_m$ and $Y^H = Y_\alpha^H - M^{-1}Y_\beta^H(r(A) + XY_\alpha^H)$ with $M = I + Y_\beta^H X$ and

$$\begin{aligned}
K_m &= [\boldsymbol{b}, A\boldsymbol{b}, \ldots, A^{m-1}\boldsymbol{b}], \\
L_m &= [\boldsymbol{c}, (A^H + \boldsymbol{cb}^H)\boldsymbol{c}, \ldots, (A^H + \boldsymbol{cb}^H)^{m-1}\boldsymbol{c}], \\
Y_\alpha &= L_m H(\alpha)^H, \\
Y_\beta &= L_m H(\beta)^H.
\end{aligned}$$

and the Hankel matrices

$$H(\alpha) = \left[\begin{array}{ccccccc}
\alpha_1 & \alpha_2 & \cdots & \alpha_{m_p} & 0 & \cdots & 0 \\
\alpha_2 & & \iddots & \iddots & & \iddots & \\
\vdots & \iddots & \iddots & & \iddots & & \\
\alpha_{m_p} & \iddots & & \iddots & & & \\
0 & & \iddots & & & & \\
\vdots & \iddots & & & & & \\
0 & & & & & & \mbox{\Huge 0}
\end{array}\right] \in \mathbb{C}^{m \times m}$$

and $H(\beta) \in \mathbb{C}^{m \times m}$ defined analogously.

Another example is [94, Theorem 1.35], which concerns updates of functions of the scaled identity matrix: Given $B, C \in \mathbb{C}^{n \times r}$, $r < n$, if f is defined on the spectrum of $\alpha I_n + BC^H$ and $C^H B$ is nonsingular,

$$f(\alpha I_n + BC^H) - f(\alpha)I_n = B(C^H B)^{-1}\left(f(\alpha I_r + C^H B) - f(\alpha)I_r\right)C^H.$$

In other, more general cases, there are no closed formulas and one has to resort to iterative algorithms for approximating $f(A + E) - f(A)$. In the works [S2] and [S3]—which are discussed in more detail below—we have proposed such algorithms for general f, based on (rational) Krylov methods. More specialized algorithms for updating the matrix square root have recently been proposed in [152] and [76].

3.2.1 B. Beckermann, D. Kressner, and M. Schweitzer, *Low-rank updates of matrix functions*, SIAM J. Matrix Anal. Appl., 39 (2018)

A basic polynomial Krylov algorithm for approximating $f(A + E) - f(A)$ is proposed in [S3], specifically for the rank-one case $E = \boldsymbol{bc}^H$. It can be extended to more general low-rank matrices E either by employing block Krylov methods (see also Section 3.2.2 below) or by considering a rank-r update as a sequence of rank-one updates. Similar to the approach outlined for Fréchet derivatives in Section 3.1.1, the general idea is to find an approximation of the form

$$f(A+\boldsymbol{b}\boldsymbol{c}^H)-f(A)\approx F_m:=U_mX_m(f)V_m^H, \tag{3.19}$$

where U_m, V_m are bases of the Krylov subspaces $\mathcal{K}_m(A,\boldsymbol{b})$ and $\mathcal{K}_m(A^H,\boldsymbol{c})$, respectively, and we have the Arnoldi relations

$$\begin{aligned} AU_m &= U_mG_m+g_{m+1,m}\boldsymbol{v}_{m+1}\boldsymbol{e}_m^H, \\ A^HV_m &= V_mH_m+h_{m+1,m}\boldsymbol{w}_{m+1}\boldsymbol{e}_m^H. \end{aligned}$$

The core factor $X_m(f)$[42] can be obtained from

$$f\left(\begin{bmatrix} G_m & \|\boldsymbol{b}\|\|\boldsymbol{c}\|\boldsymbol{e}_1\boldsymbol{e}_1^H \\ 0 & H_m^H+\|\boldsymbol{c}\|U_m^H\boldsymbol{b}\boldsymbol{e}_1^H \end{bmatrix}\right)=\begin{bmatrix} f(G_m) & X_m(f) \\ 0 & f(H_m^H+\|\boldsymbol{c}\|U_m^H\boldsymbol{b}\boldsymbol{e}_1^H) \end{bmatrix}. \tag{3.20}$$

This is motivated by [S3, Lemma 2.2], which states that

$$f\left(\begin{bmatrix} A & \boldsymbol{b}\boldsymbol{c}^H \\ 0 & A+\boldsymbol{b}\boldsymbol{c}^H \end{bmatrix}\right)=\begin{bmatrix} f(A) & f(A+\boldsymbol{b}\boldsymbol{c}^H)-f(A) \\ 0 & f(A+\boldsymbol{b}\boldsymbol{c}^H) \end{bmatrix}. \tag{3.21}$$

Note that the proof of this result is based on the integral representation (3.16). Projecting relation (3.21) onto the tensorized Krylov space $\mathcal{K}_m(A^H,\boldsymbol{c})\otimes\mathcal{K}_m(A,\boldsymbol{b})$ then exactly gives (3.20).

As an important element in the convergence analysis of the resulting method, as well as for motivating that (3.19) is indeed a sensible approximation, we derive a polynomial exactness property in [S3, Theorem 3.2].

Theorem 3.4. (Theorem 3.2 in [S3]) *Let $A\in\mathbb{C}^{n\times n}$, $\boldsymbol{b},\boldsymbol{c}\in\mathbb{C}^n$. Then the Krylov subspace approximation (3.19) is exact for all $p\in\Pi_m$, i.e.,*

$$p(A+\boldsymbol{b}\boldsymbol{c}^H)-p(A)=U_mX_m(p)V_m^H.$$

Using Theorem 3.4, it is rather straightforward to obtain a convergence result for (3.19) in the Hermitian case, exploiting a relation to polynomial approximation problems. Specifically, [S3, Theorem 4.1] states that when A is Hermitian and f is defined on a compact convex set $\mathbb{E}$ containing $W(A)\cup W(A+\boldsymbol{b}\boldsymbol{b}^H)$, the error of (3.19) satisfies

$$\|f(A+\boldsymbol{b}\boldsymbol{b}^H)-f(A)-U_mX_m(f)U_m^H\|\leq 4\min_{p\in\Pi_m}\|f-p\|_{\mathbb{E}}$$

[42] We explicitly denote the dependence on f for later utility.

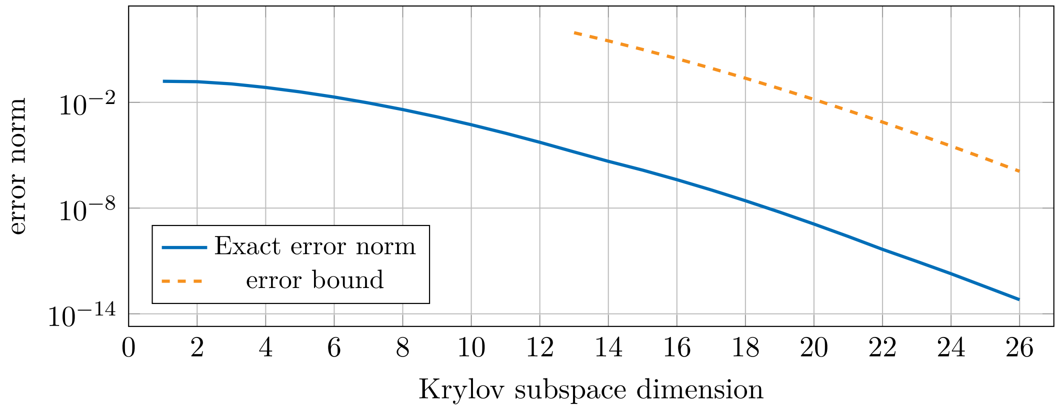


Figure 3.6: Exact error norm and error bound when updating the matrix exponential under a rank-one perturbation using the algorithm from [S3]. In this example $A \in \mathbb{R}^{100\times 100}$ is diagonal with eigenvalues equidistantly spaced in $[-20, 0]$ and $\boldsymbol{b}$ is a random vector of unit norm, resulting in $\operatorname{spec}(A - \boldsymbol{bb}^*) \subseteq [-20.2, 0] =: \mathbb{E}$. *Originally [S3, Figure 5.1].*

with the supremum norm $\|f\|_{\mathbb{E}} := \sup_{z\in\mathbb{E}} |f(z)|$. We also derive a generalization of this result to the non-Hermitian case based on the Crouzeix–Palencia theorem [49, 51], but it is of limited practical use, as it requires bounding the error of a polynomial approximation of f on a set that contains $W(\mathcal{A})$, where

$$\mathcal{A} := \begin{bmatrix} A & \boldsymbol{bc}^H \\ 0 & A + \boldsymbol{bc}^H \end{bmatrix} \tag{3.22}$$

is the block matrix from the left-hand side of (3.21). Unfortunately, as we illustrate in [S3, Section 6.2], $W(\mathcal{A})$ can be significantly larger than $W(A) \cup W(A + \boldsymbol{bc}^H)$.

In later work [S2] (see Section 3.2.2 below), we present a refinement of this result which gives much better convergence bounds also in the non-Hermitian case, based on a bivariate extension of the Crouzeix–Palencia theorem [50]. As this tool was not available at the time of writing [S3], we resorted to other techniques to still obtain convergence bounds for non-Hermitian matrices.

To be specific, we exploit the integral representation (3.16) together with techniques based on conformal mappings in the complex plane. This way, one can obtain very general bounds (see, e.g., [S3, Theorem 5.1]), which depend on non-explicit constants, though. Explicit bounds can be obtained when restricting to specific functions. In particular, we obtain superlinear convergence bounds for the matrix exponential [S3, Corollaries 5.3 and 5.5] and linear convergence bounds for Stieltjes functions [S3, Theorem 5.7, Corollaries 5.8 and 5.9]. As an example, Figure 3.6 illustrates the error bound that we obtain for the exponential; see [S3, Example 5.4] for details. We note that the convergence rate is predicted very accurately, but that the magnitude of the error is severely overestimated due to large constants in our bounds.[43]

In numerical experiments focusing on updating communicability measures subject to changes in the edges of a network, we illustrate that our method works efficiently and can outperform alternative approaches. We also present an experiment based on a convection-diffusion problem in order to illustrate the performance of our method in the non-Hermitian case. While we only have much looser convergence estimates available in this case, the practical performance still turns out to be satisfactory; see [S3, Section 6].

3.2.2 B. Beckermann, A. Cortinovis, D. Kressner, and M. Schweitzer, *Low-rank updates of matrix functions II: Rational Krylov methods*, SIAM J. Numer. Anal., 59 (2021)

Our work [S2] is a direct follow-up to [S3] discussed in Section 3.2.1 above. As new contributions to the updating algorithm for general f, we consider projection onto rational Krylov subspaces and directly incorporate a block Krylov framework. We therefore write the low-rank update as BC^H with $B, C \in \mathbb{C}^{n\times\ell}$ in the following. From an algorithmic point of view, these modifications are rather straightforward, building upon the large body of available work on rational Krylov methods; see, e.g., [29,30,89,90,142,143] and the references therein. We summarize the resulting method in Algorithm 1, where $q_m(z) = (z-\xi_1)\cdots(z-\xi_m)$ denotes the nodal polynomial corresponding to the poles $\xi_1, \dots, \xi_m \in \mathbb{C}$ of the rational Krylov space.[44] Note that due to the block setting, $U_m, V_m \in \mathbb{C}^{n\times m\ell}$ and $G_m, H_m \in \mathbb{C}^{m\ell\times m\ell}$.

Algorithm 1 Rational Krylov subspace approximation of $f(A + BC^H) - f(A)$

```
Perform $m$ steps of block rational Arnoldi to compute an orthonormal basis $U_m$ of $q_m(A)^{-1}\mathcal{K}_m(A, B)$ and set $G_m = U_m^H A U_m$.
Perform $m$ steps of block rational Arnoldi to compute an orthonormal basis $V_m$ of $\bar{q}_m(A^H)^{-1}\mathcal{K}_m(A^H, C)$ and set $H = V_m^H A^H V_m$.
Compute matrix function $F_m = f\left(\begin{bmatrix} G_m & (U_m^H B)(V_m^H C)^H \\ 0 & H_m^H + (V_m^H B)(V_m^H C)^H \end{bmatrix}\right)$.
Set $X_m(f) = F_m(1 : m\ell, m\ell + 1 : 2m\ell)$.
Return $U_m X(f) V_m^H$.
```

[43] This is a very typical phenomenon in convergence bounds for Krylov subspace methods and not a specific shortcoming of our methodology.

[44] Compared to the presentation in Section 1.4, we make the non-standard choice of a denominator polynomial from Π_m instead of Π_{m-1} here, as otherwise the resulting method would not be equivalent to the approach from [31] when applied to a rational function.

Similar to the polynomial exactness property stated in Theorem 3.4, we derive a rational exactness property as backbone for further analysis of the method [S2, Theorem 3.3]: Taking a rational function of the form $r = p_m/q_m$ with $p_m \in \Pi_m$, we have that

$$r(A + BC^H) - r(A) = U_m X_m(r) V_m^H,$$

provided that $r(A)$, $r(A + BC^H)$ as well as $r(G_m)$, $r(H_m^H + (V_m^H B)(V_m^H C)^H)$ are well-defined.

We also analyze how our method relates to the update formula from [31, Theorem 3] given in (3.18). It turns out that both approaches are mathematically equivalent in exact arithmetic when applied to a rational function, but that our new approach can be expected to be numerically more stable. Additionally, for a general function f, compared to first approximating $r \approx f$ and then using [31, Theorem 3], applying our rational Krylov method requires significantly less knowledge of spectral information on A and $A+BC^H$: Only the denominator polynomial q_m needs to be chosen "by hand", while the numerator polynomial is determined automatically by the method.

We again analyze the convergence of the method based on integral representations and conformal mappings, using similar tools as outlined in Section 3.2.1. Additionally, as was already briefly mentioned above, we also derive a strengthened convergence result for the polynomial Krylov method in the non-Hermitian case [S2, Theorem 4.3]. It relates the approximation error to polynomial approximations of the derivative of f. To be more precise,

$$\|f(A + BC^H) - f(A) - U_m X_m(f) V_m^H\|_F \leq 2(1 + \sqrt{2})^2 \|BC^H\|_F \inf_{p \in \Pi_{m-1}} \|f' - p\|_{\mathbb{E}},$$

where $\mathbb{E}$ is a compact convex set containing $W(A)$ and $W(A + BC^H)$, but *not* necessarily $W(\mathcal{A})$, with $\mathcal{A}$ from (3.22).

As an additional algorithmic contribution, we derive a specialized method for low-rank updates of the matrix sign function. When approximating $\operatorname{sign}(A)\boldsymbol{b}$ for Hermitian, nonsingular A by Krylov subspace methods, it is common practice to exploit the relation

$$\operatorname{sign}(A) = (A^2)^{-1/2} A$$

and work with the Krylov subspace $\mathcal{K}_m(A^2, A\boldsymbol{b})$ instead of $\mathcal{K}_m(A, \boldsymbol{b})$; see, e.g., [68]. This has the advantage that $W(A^2)$ does not contain a singularity of the inverse square root, so that all Krylov iterates are guaranteed to be defined. Additionally, this approach typically smooths out convergence which tends to be quite oscillatory when working with $\mathcal{K}_m(A, \boldsymbol{b})$.

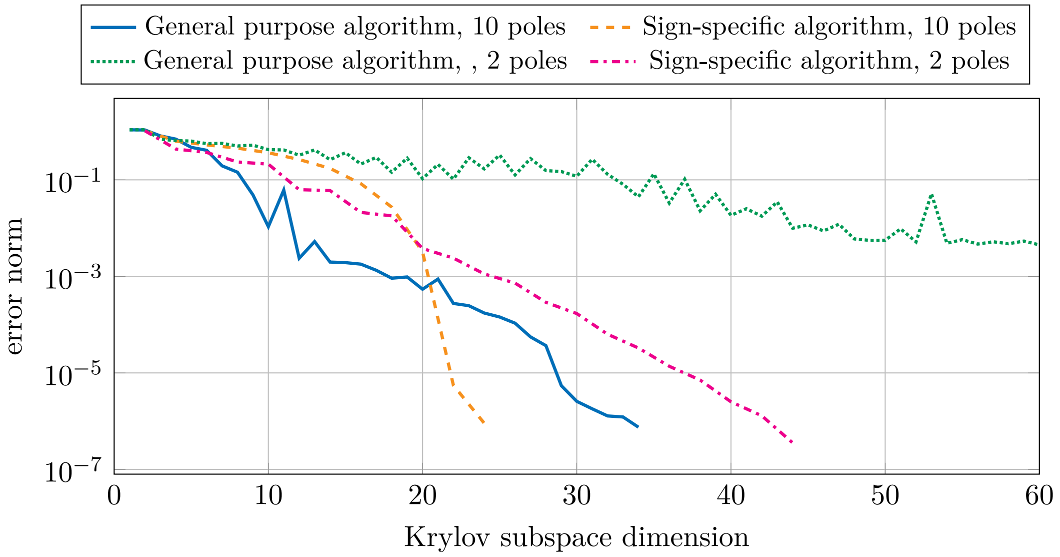


Figure 3.7: Convergence curves of general purpose and sign-specific rational Krylov methods for approximating $\operatorname{sign}(A+BB^*)-\operatorname{sign}(A)$, where $\operatorname{spec}(A) \subseteq [-1,-10^{-2}]\cup[10^{-2},1]$, $\|B\|_F = 1$. The rational Krylov methods uses the poles of a Zolotarev approximation [172] of degree 2 or 10, repeated cyclically. *Originally [S2, Figure 3].*

Adapting this approach to Hermitian low-rank updates BJB^H (with $J = J^H$) requires some special care. We have

$$\begin{aligned}&(A+BJB^H)((A+BJB^H)^2)^{-1/2} - A(A^2)^{-1/2}\\ &\quad = \quad (A+BJB^H)\big((A^2+\widetilde{D})^{-1/2} - (A^2)^{-1/2}\big) + BJB^H(A^2)^{-1/2}\end{aligned}$$

with $\widetilde{D} := ABJB^H + BJB^H(A+BJB^H)$, so that a rank-$\ell$ update of $\operatorname{sign}(A)$ corresponds to performing a rank-2ℓ update of $(A^2)^{-1/2}A$ and evaluating the action of $(A^2)^{-1/2}$ on the block vector B. In [S2, Algorithm 4], we present a rational Krylov method specifically tailored for this task, employing projection onto $q_m(A^2)^{-1}\mathcal{K}_m(A^2,[B,AB])$, which is well-suited for both tasks mentioned above. Convergence of the algorithm is proven in [S2, Theorem 5.2], with the convergence rate based on the error of best rational approximation of the inverse square root $z^{-1/2}$ on the interval

$$\mathbb{E} = \big[\min\{\lambda_{\min}(A^2), \lambda_{\min}((A+BJB^H)^2)\}, \max\{\lambda_{\max}(A^2), \lambda_{\max}((A+BJB^H)^2)\}\big].$$

We also point out a curious connection to Krylov subspace methods for Sylvester matrix equations in [S2, Section 5.2]. The solution of these can be written in terms of the matrix sign function, and it turns out that our update algorithm reduces exactly to well-known rational Krylov methods for matrix equations when appropriately applied in this setting.

3.3 Further comments on the connection between the two topics of this chapter

We conclude this chapter by providing some additional comments on a further interconnection of its two main topics—Fréchet derivatives and low-rank updates—that seems to have gone unnoticed so far. We focus specifically on the striking algorithmic similarities of the Krylov methods we presented for both cases.

Going back to the basic block representation (3.2) for the Fréchet derivative, we can alternatively consider the problem of finding the Fréchet derivative with rank-one direction term $E = \boldsymbol{b}\boldsymbol{c}^H$ as performing an update of the block-diagonal matrix

$$\mathcal{A} := \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix}.$$

Specifically, evaluating the Fréchet derivative is equivalent to performing a low-rank update from $f(\mathcal{A})$ to

$$f\left(\mathcal{A} + \begin{bmatrix} \boldsymbol{b} \\ \mathbf{0} \end{bmatrix} [\mathbf{0}, \boldsymbol{c}^H]\right) = f\left(\begin{bmatrix} A & \boldsymbol{b}\boldsymbol{c}^H \\ 0 & A \end{bmatrix}\right)$$

and then extracting the top-right block. Applying, e.g., the Krylov method from [S3] to perform this update requires building bases $\mathcal{U}_m, \mathcal{V}_m$ of the Krylov spaces

$$\mathcal{K}_m\left(\mathcal{A}, \begin{bmatrix} \boldsymbol{b} \\ \mathbf{0} \end{bmatrix}\right) \text{ and } \mathcal{K}_m\left(\mathcal{A}^H, \begin{bmatrix} \mathbf{0} \\ \boldsymbol{c} \end{bmatrix}\right)$$

and forming the corresponding projections $\mathcal{G}_m := \mathcal{U}_m^H \mathcal{A} \mathcal{U}_m$, $\mathcal{H}_m := \mathcal{V}_m^H \mathcal{A}^H \mathcal{V}_m$. Due to the block-diagonal structure of $\mathcal{A}$ and the nonzero pattern of the involved vectors, the orthonormal bases obtained from the Arnoldi method are simply given by

$$\mathcal{U}_m := \begin{bmatrix} U_m \\ 0 \end{bmatrix} \text{ and } \mathcal{V}_m := \begin{bmatrix} 0 \\ V_m \end{bmatrix}$$

and the corresponding projections of $\mathcal{A}$ are

$$\begin{aligned} \mathcal{G}_m &= [U_m^H, 0] \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix} \begin{bmatrix} U_m \\ 0 \end{bmatrix} = G_m \text{ and} \\ \mathcal{H}_m &= [0, V_m^H] \begin{bmatrix} A^H & 0 \\ 0 & A^H \end{bmatrix} \begin{bmatrix} 0 \\ V_m \end{bmatrix} = H_m. \end{aligned}$$

For obtaining the core factor, one therefore needs to evaluate

$$f\begin{pmatrix} G_m & \|\boldsymbol{b}\|\|\boldsymbol{c}\|\boldsymbol{e}_1\boldsymbol{e}_1^H \\ 0 & H_m^H + \|\boldsymbol{c}\|\mathcal{V}_m^H \begin{bmatrix} \boldsymbol{b} \\ \mathbf{0} \end{bmatrix} [\mathbf{e}_1^H, \mathbf{0}^H] \end{pmatrix} = f\begin{pmatrix} G_m & \|\boldsymbol{b}\|\|\boldsymbol{c}\|\boldsymbol{e}_1\boldsymbol{e}_1^H \\ 0 & H_m^H \end{pmatrix},$$

where the equality follows from $\mathcal{V}_m^H \begin{bmatrix} \boldsymbol{b} \\ \mathbf{0} \end{bmatrix} = \mathbf{0}$. This is precisely (3.6), so that both approaches lead to exactly the same algorithm. An advantage of the approach in [S11] is that it allows working just with $W(A)$ for the convergence analysis, while the "detour" via more general low-rank updates would require to also incorporate $W(A + \boldsymbol{b}\boldsymbol{c}^H)$ into the analysis.

CHAPTER 4

DECAY BOUNDS AND PROBING METHODS

The matrix function $f(A)$ is in general a full matrix, even when $A \in \mathbb{C}^{n \times n}$ is sparse. However, many of its entries are typically extremely small, the more so the farther they are away from the sparsity pattern of A.[45] See Figure 4.1 for an example of this phenomenon.

This decay behavior in matrix functions has been studied since a long time. Early publications on the topic mostly focused on the important special case of banded A, and mostly considered the inverse [57, 58, 67, 78, 79, 110, 130] and the exponential [18, 23, 106, 122, 139]. Further results on other classes of functions and for more general sparsity patterns can be found in, e.g., [16, 20, 21, 23, 139, 148]; see also the survey [15].

It is of great interest to be able to accurately predict the decay in matrix functions, as this, e.g., allows to construct accurate *sparse approximations* of $f(A)$ [138] or related quantities; see, e.g., [34–36, 88] for applications in Markov chain queuing models, quantum dynamics and inverse covariance estimation. More generally, if a rapid decay in $f(A)$ is present, this can be exploited for designing linearly scaling algorithms for a wide variety of computational problems involving matrix functions [20, 43].

Another application which heavily relies on decay in $f(A)$ that has emerged in recent years is trace estimation via probing methods [22, 84, 118, 158, 159, 161], which we cover in more detail in Section 4.2 below.

[45] How exactly this should be understood will be made more precise in Section 4.1 below.

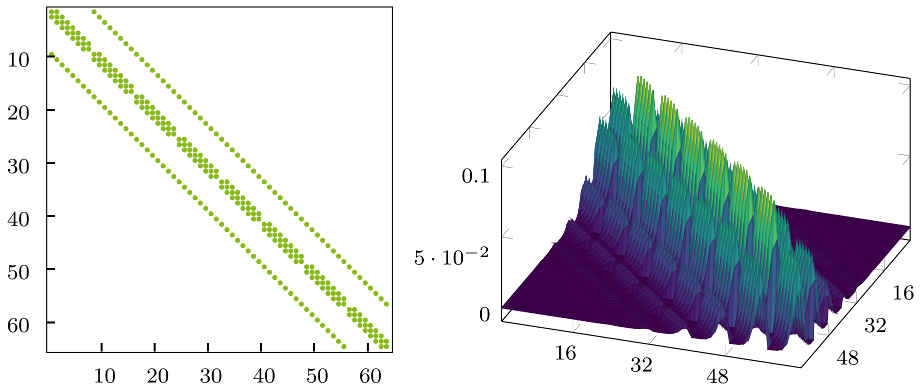


Figure 4.1: Exponential of discretized two-dimensional Laplace operator $A \in \mathbb{R}^{64\times 64}$. *Left:* Sparsity pattern of A. *Right:* Magnitude of the entries in $\exp(-A)$.

4.1 Decay bounds for matrix functions

A standard approach for obtaining decay bounds for functions of Hermitian—or more generally normal—matrices which dates back at least to [58] is to exploit a strong relationship to polynomial approximation. In order to explain this approach, we first define the graph of A and the geodesic distance in that graph.

Definition 4.1. Let $A \in \mathbb{C}^{n\times n}$. The (possibly directed) *graph of A* is $G_A = (\mathcal{V}_A, \mathcal{E}_A)$, with nodes $\mathcal{V}_A := \{1, \ldots, n\}$ and edges $\mathcal{E}_A := \{(i,j) : a_{ij} \neq 0, i \neq j\}$.

The *geodesic distance* between two nodes i and j in G_A, denoted as $\operatorname{dist}(i,j)$, is the length of the shortest path starting at node i and ending at node j.

It is well-known that the nonzero entries in polynomials of a sparse matrix A “spread out” along paths in G_A. To be specific, if $p_m \in \Pi_m$, we have

$$[p_m(A)]_{ij} = 0 \text{ whenever } \operatorname{dist}(i,j) > m.\text{[46]} \tag{4.1}$$

Using this relation, setting $m := \operatorname{dist}(i,j) + 1$ and letting p_m be *any* polynomial of degree m, we have for normal A,

$$|[f(A)]_{ij}| = |[f(A)]_{ij} - [p_m(A)]_{ij}| \leq \|f(A) - p_m(A)\| \leq \max_{z\in\mathbb{E}} |f(z) - p_m(z)|,$$

where $\mathbb{E} \subset \mathbb{C}$ is some set containing $\operatorname{spec}(A)$. In particular, we thus have

$$|[f(A)]_{ij}| \leq \min_{p_m^*\in\Pi_m} \max_{z\in\mathbb{E}} |f(z) - p_m^*(z)|,$$

[46] See also Example 1.4 for further explanations on the origin of this relation.

i.e., the magnitude of the entries in $f(A)$ is directly related to the error of best polynomial approximation of f on the set $\mathbb{E}$. Note that when A is Hermitian, $\mathbb{E}$ can be taken as an interval on the real line, while for normal—but not Hermitian—A one needs to take a more general convex set in the complex plane.

As the error of best polynomial approximation decreases monotonically with m, this result also suggests that the magnitude of $[f(A)]_{ij}$ can be expected to become smaller the larger the geodesic distance $\operatorname{dist}(i,j)$ is.[47]

4.1.1 A. Frommer, C. Schimmel, and M. Schweitzer, *Bounds for the decay of the entries in inverses and Cauchy–Stieltjes functions of certain sparse, normal matrices*, Numer. Linear Algebra Appl., 25 (2018)

In our work [S6], we mostly focused on deriving bounds for the decay in inverses of normal matrices with spectrum located in a line segment $[\lambda_1, \lambda_2]$ in the complex plane. This class of matrices is more general than the class of Hermitian matrices, but easier to handle than general normal matrices. As important, practically relevant special cases, it contains (shifted) skew-Hermitian matrices, and Hermitian matrices shifted by a complex multiple of the identity.

Our main result [S6, Theorem 2] concerning this type of matrices asserts that if $0 \notin [\lambda_1, \lambda_2]$, then for $i \neq j$

$$|[A^{-1}]_{ij}| \leq 2\,\|A^{-1}\|\,\frac{1}{q^{\operatorname{dist}(i,j)-1}} \tag{4.2}$$

with

$$q = e^{\operatorname{Re}(\operatorname{arcosh}(x))} > 1 \text{ and } x = \frac{\lambda_1+\lambda_2}{\lambda_2-\lambda_1},$$

which is proven by exploiting approximation properties of Chebyshev polynomials. The result can be specialized to the cases mentioned above, potentially giving more explicit decay bounds.

When the spectrum of A contains a "hole" (typically around the intersection of the line segment $[\lambda_1, \lambda_2]$ with the real axis), it can be beneficial to reflect this in the decay bounds instead of considering just a single line segment containing the spectrum. For example, let $A = aI + S$ where $0 \neq a \in \mathbb{R}$ and S is a

[47] Note that some of our publications discussed in the following originally stated their results for banded matrices. For a more consistent presentation, we adapt these results to the geodesic distance (except for results in Section 4.1.2 which only apply to tridiagonal matrices).

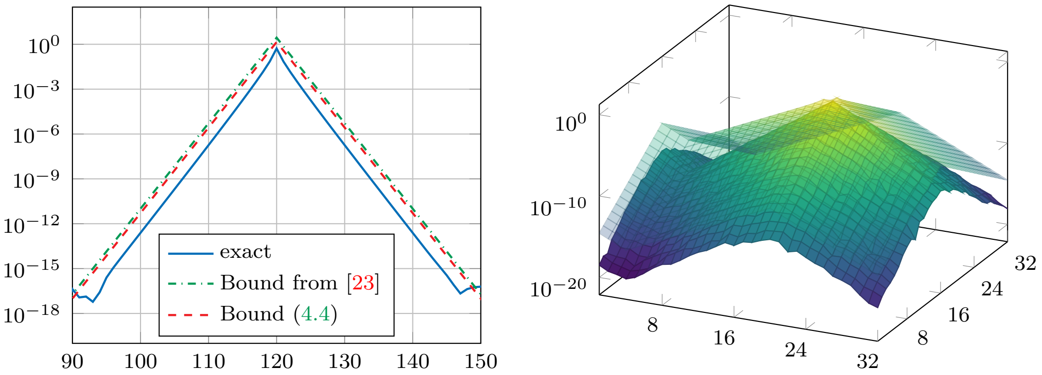


Figure 4.2: *Left:* Bounds for $[A^{-1/2}]_{ij}$, $j = 120$ for the matrix $A = \text{tridiag}(-1, 4, -1)$, $n = 200$. *Right:* Bound (4.3) for $(sI + D)^{-1}_{ij}$, $n = 1024$, $j = 504$ for a staggered Schwinger discretization. *Adapted from [S6, Figures 5 & 8].*

nonsingular, skew-Hermitian matrix with spectrum contained in a set of the form $i[-b_2, -b_1] \cup i[b_1, b_2]$, where $b_1 = \min_{\lambda \in \text{spec}(S)} |\lambda|$ and $b_2 = \max_{\lambda \in \text{spec}(S)} |\lambda|$. Then

$$|[A^{-1}]_{ij}| \leq 2 \, \|A^{-1}\| \cdot \begin{cases} q^{\text{dist}(i,j)} & \text{for } \text{dist}(i,j) \text{ odd}, \\ q^{\text{dist}(i,j)-1} & \text{for } \text{dist}(i,j) \text{ even}, \end{cases}$$

where

$$q = \left(\sqrt{x} + \sqrt{x+1}\right)^{-1} \quad \text{with} \quad x = \frac{a^2 + b_1^2}{b_2^2 - b_1^2}. \tag{4.3}$$

In a detailed comparison, we show that (4.3) potentially predicts the slope of the actual decay a lot more accurately than (4.2), in particular when the gap around the real axis is large. We also compare to previously proposed bounds from [58,79] and are able to show that also in cases where (4.3) gives the same slope as the results of [58, 79], the constant in front of $q^{\text{dist}(i,j)}$ is much smaller, by a factor that scales with the reciprocal of the condition number of A, so that our decay bounds are much tighter for ill-conditioned A.

In addition to results for the inverse, we also consider Cauchy–Stieltjes functions of normal matrices. For Hermitian positive definite A, we improve upon a previous result of [23]: We show in [S6, Theorem 4] that when f is a Cauchy–Stieltjes function, we have

$$|[f(A)]_{ij}| \leq 2 f(\lambda_{\min}) \, q^{\text{dist}(i,j)} \text{ with } q = \frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1}. \tag{4.4}$$

This result improves over [23, Theorem 4.6] in two ways: First, the constant in the decay bound is much smaller, and second, the bound has an explicit form that can be evaluated analytically, while the bound of [23] can only be evaluated using numerical quadrature.

In [S6, Lemma 3, Theorem 5 & 6] we also consider Cauchy–Stieltjes functions of other classes of normal matrices with spectrum on a line segment, obtaining results which are reminiscent of what we obtained for the inverse, albeit with larger and more complicated (and sometimes non-explicit) constants.

In Figure 4.2, we illustrate the bounds we obtain for two model problems. On the left-hand side, we report decay bounds for the square root of a shifted one-dimensional Laplace operator, i.e., a Hermitian positive definite matrix. We observe that our bound predicts the same decay rate as the previous bound from [23], but involves a slightly smaller constant and is thus a bit sharper. We stress again that the main merit of the bound lies in the availability of a closed form, while the bounds from [23] can in general only be evaluated numerically. The right-hand side shows our new bound for the staggered Schwinger discretization from quantum electrodynamics on a periodic two-dimensional lattice.[48] This discretization yields a system

$$(sI + D)\psi = \phi,$$

where D is skew-Hermitian and has a spectrum that is symmetric with respect to the origin (due to the odd-even structure of the coupling). This is exactly the setting that is required for using our decay estimate (4.3).

4.1.2 A. Frommer, C. Schimmel, and M. Schweitzer, *Non-Toeplitz decay bounds for inverses of Hermitian positive definite tridiagonal matrices*, Electron. Trans. Numer. Anal., 48 (2018)

The bounds derived in [S6] are of the form

$$|[f(A)]_{ij}| \leq c \cdot q^{\operatorname{dist}(i,j)}.$$

In a direct follow-up work [S7], we address two shortcomings that these bounds have—and that they have in common with essentially all other decay bounds for $f(A)$ or A^{-1} that had been proposed in the literature up to that point. We restrict to the case $f(z) = z^{-1}$ and tridiagonal A, so that $\operatorname{dist}(i,j) = |i-j|$, i.e., the bounds take the form

$$|[A^{-1}]_{ij}| \leq c \cdot q^{|i-j|}. \tag{4.5}$$

[48] Note that we arranged the column entries according to the underlying lattice, as this gives a better intuition about the actual graph distances than using the linear ordering induced by the vector indices.

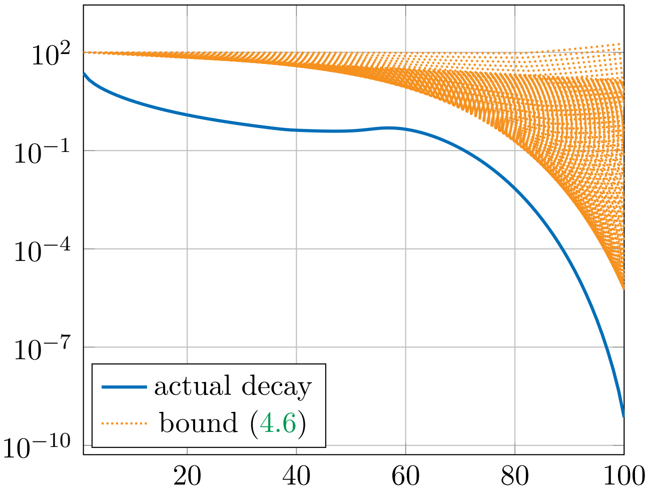


Figure 4.3: Actual magnitude of the entries of the 50th column of A^{-1}, where $A \in \mathbb{R}^{100\times 100}$ is the matrix from [S7, Example 1.2], together with bound (4.6) for $\ell = 0, \dots, 99$. *Adapted from [S7, Figure 3.2].*

The first shortcoming of (4.5) that we address is that it can only predict a *linear* decay in the entries of A^{-1}, while one easily finds (or constructs) examples in which the actual decay that one observes is superlinear, in particular when the eigenvalues of A form clusters within the spectral interval.

This phenomenon can be explained quite easily by adapting a technique that is also sometimes used for explaining superlinear convergence behavior of the conjugate gradient method [121].

This technique requires detailed information on the spectrum, though (instead of just the smallest and largest eigenvalue), so that it can only be seen as a theoretical tool, not as a practical method for obtaining decay bounds in an actual computation. It is based on the ℓth effective condition number, which is defined as

$$\kappa_\ell(A) = \frac{\lambda_{n-\ell}}{\lambda_1},$$

where $\lambda_{\min}(A) = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n = \lambda_{\max}(A)$, so that $\kappa_0(A) = \kappa(A)$ is the usual two-norm condition number of A. In [S7, Theorem 3.1], we show that the entries of A^{-1} can be bounded in terms of the effective condition number as

$$|[A^{-1}]_{ij}| \leq \frac{2}{\lambda_1} q_\ell^{|i-j|-\ell} \quad \text{where} \quad q_\ell = \frac{\sqrt{\kappa_\ell(A)}-1}{\sqrt{\kappa_\ell(A)}+1} \tag{4.6}$$

for all $\ell = 0, 1, \dots, |i-j|-1$. Equation (4.6) gives a family of bounds for each entry of A^{-1}, and the value of ℓ which gives the tightest bound depends on the specific entry and the eigenvalue distribution of A, as increasing ℓ improves the decay rate q_ℓ but reduces its exponent. See Figure 4.3 for an illustration of the family of bounds obtained this way.[49] We can observe that the lower envelope of all bounds (4.6) quite accurately predicts the actual decay behavior in the depicted column of A^{-1}.

The second problem is best observed when interpreting (4.5) as an entry-wise inequality $|A^{-1}| \leq Q$, where

$$Q = c \cdot \begin{bmatrix} q^0 & q^1 & q^2 & \cdots & q^{n-1} \\ q^1 & q^0 & q^1 & \ddots & \vdots \\ \vdots & q^1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & q^1 \\ q^{n-1} & \cdots & \cdots & q^1 & q^0 \end{bmatrix}.$$

The matrix Q is a Toeplitz matrix, which lies in the nature of the bound (as it only depends on the distance of i and j). However, there is no reason to assume that A^{-1} is a Toeplitz matrix, even when A itself is Toeplitz. Therefore, a bound of this form cannot be expected to be descriptive for all parts of A^{-1}, but rather only for the portion which shows the slowest decay, with the bounds for other entries in A necessarily following the same pattern.

Our approach to overcome this limitation (for tridiagonal matrices) is based on a splitting of A into a block diagonal matrix and a rank-one matrix

$$A = \begin{bmatrix} A_{11} & A_{22} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} + \boldsymbol{u}\boldsymbol{u}^H, \quad \boldsymbol{u} = \alpha \left(\boldsymbol{e}_k + \frac{a_{k+1,k}}{|a_{k+1,k}|} \boldsymbol{e}_{k+1} \right) \in \mathbb{C}^n, \quad (4.7)$$

where $A_{11} \in \mathbb{C}^{k \times k}, A_{22} \in \mathbb{C}^{(n-k)\times(n-k)}$ and $\alpha = \sqrt{|a_{k+1,k}|}$. We then apply the Sherman–Morrison–Woodbury formula (3.17) to the decomposition (4.7), which allows to write A^{-1} as the sum of a block diagonal and a rank-one term, which both exhibit an exponential decay, from which the final result follows. One limitation of this technique is that it requires B_1 and B_2 from (4.7) to be positive definite. This is not necessarily the case even when A is positive definite, but it can be guaranteed when A is, in addition, diagonally dominant.

In that case, according to [S7, Theorem 4.1], we can bound the entries of A^{-1} as

$$|[A^{-1}]_{ij}| \leq \begin{cases} c_1\, q_1^{|i-j|} + c_1^2\, \widetilde{c}\, q_1^{2k-j-i} & \text{for } i,j \leq k \\ c_2\, q_2^{|i-j|} + c_2^2\, \widetilde{c}\, q_2^{i+j-2(k+1)} & \text{for } i,j > k \\ c_1\, c_2\, \widetilde{c}\, q_1^{k-i}\, q_2^{j-k-1} & \text{for } i \leq k < j \\ c_1\, c_2\, \widetilde{c}\, q_1^{j-k}\, q_2^{i-k-1} & \text{for } j \leq k < i \end{cases} \qquad (4.8)$$

[49] Details on how the matrix A for this is example is constructed are given in [S7, Example 1.2], following the principles outlined in [117, Section 6.1].

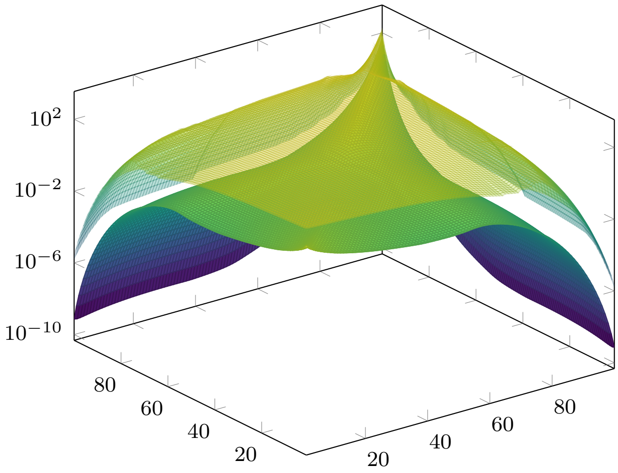


Figure 4.4: Actual magnitude of the entries of A^{-1}, where $A \in \mathbb{R}^{100\times 100}$ is the matrix from [S7, Example 1.2], together with bounds obtained by combining (4.8) with (4.6) for best possible ℓ. *Originally [S7, Figure 4.2].*

where

$$q_s = \frac{\sqrt{\kappa(B_s)}-1}{\sqrt{\kappa(B_s)+1}}, \quad \kappa(B_s) = \frac{\lambda_{\max}(B_s)}{\lambda_{\min}(B_s)}, \quad c_s = \frac{2}{\lambda_{\min}(B_s)}, \quad s = 1,2$$

and

$$\widetilde{c} = \frac{|a_{k+1,k}|}{1 + |a_{k+1,k}| \left(\frac{1}{\lambda_{\max}(B_1)} + \frac{1}{\lambda_{\max}(B_2)} \right)}.$$

Note that the bounds in (4.8) can also be combined with the techniques using the effective condition number discussed above, but we refrain from explicitly stating the resulting bounds (as we also did in [S7]) as the technique is straightforward but notation gets heavy and cluttered.

Figure 4.4 shows the bounds obtained by combining (4.6) and (4.8) in comparison to the classical (Toeplitz-structured) bounds from [58], again for the matrix from [S7, Example 1.2]. One can observe that our bound predicts the qualitative decay behavior—which is quite different in different columns of the matrix—very well, although the size of the entries is overestimated by two or three orders of magnitude.

We want to remark that our results from [S7]—both regarding superlinear bounds as well as non-Toeplitz bounds—were recently extended and generalized to the matrix sign function and spectral projectors in [21].

4.1.3 M. Schweitzer, *Decay bounds for Bernstein functions of Hermitian matrices with applications to the fractional graph Laplacian*, Electron. Trans. Numer. Anal., 55 (2022)

In [S13], we derive decay bounds for Bernstein functions (see Definition 1.10), which require slightly different techniques than Cauchy–Stieltjes functions. An added difficulty arises from the fact that our main motivation for this work was studying the *fractional graph Laplacian* L_G^α, where $\alpha \in (0,1)$ [25, 26, 32, 33, 72] and L_G is the Laplacian of an undirected graph G, which is a singular, positive semidefinite matrix; cf. Example 1.5. The lack of analyticity of z^α at the origin requires special care and ultimately also leads to much slower decay than what one observes for analytic functions.

Previously, decay bounds for the fractional graph Laplacian had been found in [16, 33], which predict a power-law decay away from the sparsity pattern of L_G. In particular, based on Jackson's theorem [128], it is shown in [16, Corollary 3.1] that if $\operatorname{dist}(i,j) \geq 2$, we have

$$|[L_G^\alpha]_{ij}| \leq \left(1 + \pi^2/2\right) \cdot \left(\frac{\rho(L_G)}{2}\right)^\alpha \cdot (\operatorname{dist}(i,j) - 1)^{-\alpha}, \tag{4.9}$$

where $\rho(L_G)$ denotes the spectral radius of L_G.

Using the results outlined in the following, we obtain that the exponent in (4.9) can be improved—at least asymptotically— from $-\alpha$ to -2α. Our results are based on exploiting the intimate relation between Bernstein functions and the exponential, given by the Lévy–Khintchine representation (1.11), which allows one to essentially integrate over decay bounds for the exponential in order to obtain a decay bound for a Bernstein function. In particular, we use the result of [23, Theorem 4.2], which is in turn based on the fundamental convergence result for Lanczos approximations of the exponential due to Hochbruck and Lubich [99, Theorem 2]. Using these results gives the following general bound [S13, Lemma 3.2] which—as the result in (4.9)—is valid for all i,j with $\operatorname{dist}(i,j) \geq 2$,

$$\begin{aligned} |[f(A)]_{ij}| \quad \leq \quad & 10 \int_0^{\frac{2\operatorname{dist}(i,j)}{\rho(A)}} \frac{4\exp(-\frac{1}{4}\rho(A)t)}{\rho(A)t} \left(\frac{e\rho(A)t}{4\operatorname{dist}(i,j)}\right)^{\operatorname{dist}(i,j)} \mathrm{d}\lambda(t) \\ & +10 \int_{\frac{2\operatorname{dist}(i,j)}{\rho(A)}}^{\frac{\operatorname{dist}(i,j)^2}{\rho(A)}} \exp\left(-\frac{4\operatorname{dist}(i,j)^2}{5\rho(A)t}\right) \mathrm{d}\lambda(t) \\ & + \int_{\frac{\operatorname{dist}(i,j)^2}{\rho(A)}}^{\infty} |[\exp(-tA)]_{ij}| \; \mathrm{d}\lambda(t). \end{aligned} \tag{4.10}$$

The bound (4.10) can be made more explicit for particular Bernstein functions. In [S13, Theorem 3.3], we apply the result to fractional powers, giving

$$\begin{aligned}
|[A^\alpha]_{ij}| \quad \leq \quad & \frac{\alpha}{\Gamma(1-\alpha)} \cdot \left(\frac{10e^{\mathrm{dist}(i,j)}\rho(A)^\alpha}{4^\alpha \mathrm{dist}(i,j)^{\mathrm{dist}(i,j)}} \cdot \gamma\left(\mathrm{dist}(i,j)-\alpha-1, \frac{\mathrm{dist}(i,j)}{2}\right) \right. \\
& +10\left(\frac{5\rho(A)}{4\mathrm{dist}(i,j)^2}\right)^\alpha \cdot \left(\Gamma\left(\alpha, \frac{4}{5}\right) - \Gamma\left(\alpha, \frac{2\mathrm{dist}(i,j)}{5}\right)\right) \\
& \left. + \frac{\rho(A)^\alpha}{\alpha \cdot \mathrm{dist}(i,j)^{2\alpha}} \right), \qquad (4.11)
\end{aligned}$$

where $\Gamma(z,s)$ and $\gamma(z,s)$ denote the *upper and lower incomplete Gamma function*,

$$\Gamma(z,s) = \int_s^\infty t^{z-1}e^{-t}\,\mathrm{d}t \quad \text{and} \quad \gamma(z,s) = \int_0^s t^{z-1}e^{-t}\,\mathrm{d}t,$$

respectively.

Returning to the initial motivation of studying strength of connection in the fractional graph Laplacian, it is important to observe that the first term in the above bound goes to zero exponentially when $\mathrm{dist}(i,j)$ increases, so that asymptotically, the second and third term control the decay behavior in L_G^α. Thus, [S13, Theorem 3.3] gives an asymptotic estimate

$$|[L_G^\alpha]_{ij}| \lesssim C \cdot \mathrm{dist}(i,j)^{-2\alpha},$$

where C is a constant that is independent of i and j, thereby improving the exponent in the power-law decay compared to what was obtained in [16, 33]. An illustration of this improvement is given in Figure 4.5 for a random geometric graph; see [S13, Example 4.4] for details on how exactly the problem is set up.

Based on the results obtained for the semidefinite case, we also derive decay bounds for Bernstein functions of positive definite matrices, exploiting the well-known relation $\exp(A-\lambda I) = \exp(-\lambda)\exp(A)$. This allows to shift the smallest eigenvalue $\lambda_{\min}$ of a positive definite matrix A to zero before applying [23, Theorem 4.2], yielding an additional factor $\exp(-\lambda_{\min}t)$ in all integrals. This way, we can obtain improved, sharper bounds, which, however—in contrast to the semidefinite case—cannot be cast into a closed form for fractional powers[50] and instead have to be evaluated by numerical quadrature. This, however, typically does not pose any problem for general-purpose integration packages. The improved bound

[50] To be precise, only the second integral in (4.12) does not have a closed form solution.

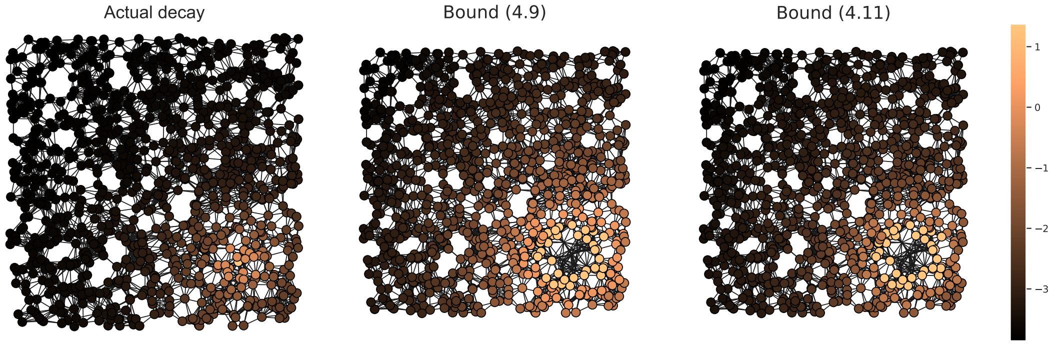


Figure 4.5: Decay in one column of the fractional Laplacian $L_G^{1/2}$ for a random geometric graph G (on a logarithmic scale). *Left:* Actual decay, *Center:* Bound (4.9) based on Jackson's theorem, *Right:* New bound (4.11). In the center and right graph, nodes for which no bound is available are not drawn. *Originally [S13, Figure 4.2].*

is again valid for $\operatorname{dist}(i,j) \geq 2$ and reads

$$\begin{aligned} |[f(A)]_{ij}| \quad \leq \quad & 10 \int_0^{\frac{\operatorname{dist}(i,j)}{2\sigma}} \frac{\exp(-(\sigma+\lambda_{\min})t)}{\sigma t} \left(\frac{e\sigma t}{\operatorname{dist}(i,j)}\right)^{\operatorname{dist}(i,j)} \mathrm{d}\lambda(t) \\ & +10 \int_{\frac{\operatorname{dist}(i,j)}{2\sigma}}^{\frac{\operatorname{dist}(i,j)^2}{4\sigma}} \exp(-t\lambda_{\min}) \cdot \exp\left(-\frac{\operatorname{dist}(i,j)^2}{5\sigma t}\right) \mathrm{d}\lambda(t) \\ & + \int_{\frac{\operatorname{dist}(i,j)^2}{4\sigma}}^{\infty} \exp(-t\lambda_{\min})\, \mathrm{d}\lambda(t), , \end{aligned} \tag{4.12}$$

where $\sigma = (\lambda_{\max} - \lambda_{\min})/4$; see [S13, Lemma 3.6]. We also derive alternative decay bounds for A^α exploiting the relation $A^\alpha = A \cdot A^{\alpha-1}$ involving the Cauchy–Stieltjes function $f(z) = z^{\alpha-1}$ (see [S13, Section 3.3]), but as these bounds turned out to be always worse than (4.12), and increasingly more so the more ill-conditioned A becomes, we refrain from reporting them here.

4.1.4 M. Schweitzer, *Sensitivity of matrix function based network communicability measures: Computational methods and a priori bounds*, SIAM J. Matrix Anal. Appl., 44 (2023) – Section 5

We return to [S15], which was already partly covered in Section 3.1.2. The second part of this paper discusses decay bounds for Fréchet derivatives of matrix functions with structured direction terms, with particular focus on modifying adjacency matrices of complex networks. The results we obtain confirm the intuition that a modification of certain nodes or edges in a network typically only influences their direct surroundings heavily, but does not influence nodes which are "far away"; see also [140] for similar results in a slightly different setting.

Similar to the case of matrix functions, decay bounds for Fréchet derivatives can be obtained by exploiting polynomial approximation properties. The starting point for doing so is [S15, Lemma 5.2], which can be seen as an analogue of (4.1) for Fréchet derivatives: If $p_m \in \Pi_m$, then

$$[L_{p_m}(A, \boldsymbol{e}_i \boldsymbol{e}_j^H)]_{uv} = 0 \quad \text{if } d(u,i) + d(j,v) \geq m. \tag{4.13}$$

The direction term in the Fréchet derivative in (4.13) plays a role when investigating sensitivity with respect to changes in the edge (i,j), cf. (3.8). Additionally, in the context of removing a node v from the network the direction term $E_v = -(\boldsymbol{e}_v \boldsymbol{a}_{v:} + \boldsymbol{a}_{:v} \boldsymbol{e}_v^H)$ plays a role, where $\boldsymbol{a}_{v:}$ and $\boldsymbol{a}_{:v}$ denote the vth row and column of A, respectively. For such direction terms, a similar result holds, namely

$$[L_{p_m}(A, E_v)]_{u_1 u_2} = 0 \quad \text{if } d(u_1, v) + d(v, u_2) \geq m + 1.$$

By combining these results with the bivariate extension of the Crouzeix-Palencia theorem from [50], one can relate decay in the Fréchet derivative to polynomial approximation of f' on $W(A)$. According to [S15, Theorem 5.4 & Remark 5.5], it holds

$$|[L_f(A, E_{ij})]_{uv}| \leq C \cdot \min_{p \in \Pi_{m(u,v)-1}} \max_{z \in W(A)} |f'(z) - p(z)| \tag{4.14}$$

and

$$|[L_f(A, E_v)]_{u_1 u_2}| \leq C \cdot \sqrt{\deg(v)} \min_{p \in \Pi_{m_v(u_1,u_2)-1}} \max_{z \in W(A)} |f'(z) - p(z)| \tag{4.15}$$

where

$$C = \begin{cases} 1 & \text{if } A \text{ is normal,} \\ \left(1 + \sqrt{2}\right)^2 & \text{otherwise,} \end{cases}$$

and we denote by $\deg(v) := \sum_{u=1}^n w_{vu}$ the "weighted degree" (or "strength") of node v, and we define

$$\begin{aligned} m(u,v) &:= d(u,i) + d(j,v), \\ m_v(u_1,u_2) &:= d(u_1,v) + d(v,u_2) + 1. \end{aligned}$$

Note that for the exponential function occurring in network sensitivity, $f = f'$, so that approximation results for $\exp(z)$ can directly be employed (while for other functions f, polynomial approximation of the derivative might not always be well studied).

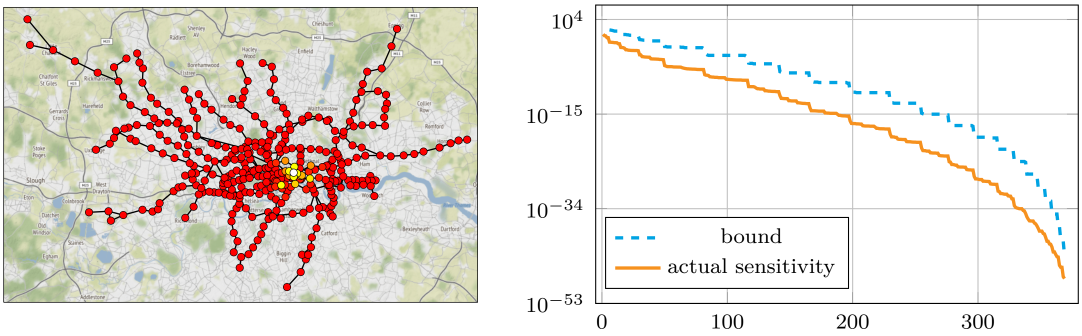


Figure 4.6: Bounds for the sensitivity of subgraph centrality with respect to removal of *Moorgate station* (depicted in white) in a network describing London public transport. *Left:* Visualization on London map. Lighter colors correspond to high sensitivity, while darker colors correspond to low sensitivity. *Right:* Plot of bounds, sorted descendingly. *Originally [S15, Figures 5.1 & 5.2]. (Street map generated with* `cartopy` *[129], map data © OpenStreetMap).*

Based on (4.14) and (4.15), several decay results can be obtained by tailoring them to the shape and location of $W(A_G)$. Specifically, we present decay results for undirected graphs, i.e., symmetric A_G, where $W(A_G)$ is a real interval [S15, Corollary 5.6 & 5.11] and for directed graphs where $W(A_G)$ is contained in a disk [S15, Corollary 5.9 & 5.12]. The former results are based on a polynomial approximation result from [S11, Lemma 2], while the latter employ conformal mappings and approximation by Faber polynomials.

As an example for this type of result, we report the precise bound from [S15, Corollary 5.11] in the following and then illustrate it on real-world data. It states that if $\sqrt{\lambda_{\max} - \lambda_{\min}} + 1 \leq m(u_1, u_2) \leq \dfrac{\lambda_{\max} - \lambda_{\min}}{2} + 1$, with m defined in (4.16), we have the bound

$$|[L_{\exp}(A_G, E_v)]_{u_1 u_2}| \leq 2\sqrt{\deg(v)} \frac{\lambda_{\max} - \lambda_{\min}}{m(u_1, u_2) - 1} e^{\lambda_{\max} - \frac{(m(u_1,u_2)-1)^2}{\frac{5}{4}(\lambda_{\max} - \lambda_{\min})}} \tag{4.16}$$

and if $m(u_1, u_2) > \dfrac{\lambda_{\max} - \lambda_{\min}}{2} + 1$, we have the bound

$$|[L_{\exp}(A_G, E_v)]_{u_1 u_2}| \leq C \cdot \left(\frac{e \cdot (\lambda_{\max} - \lambda_{\min})}{4(m(u_1, u_2) - 1) + 2(\lambda_{\max} - \lambda_{\min})} \right)^{m(u_1,u_2)-1} \tag{4.17}$$

with $C = 8\sqrt{\deg(v)} \frac{e^{\lambda_{\max}(m(u_1,u_2)-1)}}{\lambda_{\max} - \lambda_{\min}}$.

Figure 4.6 illustrates the bounds (4.16)–(4.17). We consider a network describing public transport in London [54, 55] and investigate how the subgraph sensitivity of nodes reacts to the removal of the node corresponding to *Moorgate station* near the city center (e.g., modeling an outage or a terrorist attack). One can clearly observe that the nearby stations are most strongly affected, while the effect on nodes at the outskirts of London is very small. It is also interesting to observe that the “staircase-like” shape of the bound is not only an artifact of the techniques used for obtaining the results but is actually also visible in the actual sensitivity values.

4.2 Probing methods for trace estimation

An important task in many areas of scientific computing and data science is estimating the trace of an implicitly given matrix,

$$\operatorname{tr}(B) = \sum_{i=1}^{n} [B]_{ii}.$$

Here, *implicitly given matrix* typically means that the matrix can only be accessed through a routine that returns its matrix vector product, $\boldsymbol{x} \mapsto B\boldsymbol{x}$. An important special case on which we want to focus here arises when $B = f(A)$ is the function of a large, sparse matrix A. In that case, matrix vector products $B\boldsymbol{x} = f(A)\boldsymbol{x}$ can, e.g., be approximated via a (polynomial or rational) Krylov subspace method, see Section 1.4, but $f(A)$ itself cannot be formed.

Probing methods are a special class of trace estimators which try to exploit the sparsity structure of the underlying matrix A, as this sparsity structure usually translates into a decay pattern in $f(A)$, as we discussed in Section 4.1. These methods were originally introduced in [161] and then later adapted and refined in [8, 22, 118, 158, 159].

Specifically, the first step in a probing method is to construct a partition $V_1, \ldots, V_m$ of the set $\{1, \ldots, n\}$. The most common approach for doing so is by choosing $V_1, \ldots, V_m$ according to the colors of a distance-d coloring [77, 150] of the graph of A, where d is a suitably chosen parameter.[51] This means that $i, j \in V_\ell$ implies that $\operatorname{dist}(i, j) > d$.

Given this partitioning, one defines the m probing vectors

$$\boldsymbol{v}_\ell := \sum_{i \in V_\ell} \boldsymbol{e}_i, \qquad \text{for } \ell \in \{1, \ldots, m\} \tag{4.18}$$

and the corresponding trace estimate

$$\operatorname{tr}(f(A)) \approx \mathcal{T}_m(f(A)) := \sum_{\ell=1}^{m} \boldsymbol{v}_\ell^H f(A) \boldsymbol{v}_\ell. \tag{4.19}$$

[51] Appropriate choices of d typically depend both on the accuracy that one aims for and on properties of A and f

It is straightforward to prove that the error of the estimator (4.19) satisfies

$$|\mathrm{tr}(f(A)) - \mathcal{T}_m(f(A))| = \left| \sum_{\ell=1}^{m} \sum_{\substack{i,j \in V_\ell \\ i \neq j}} [f(A)]_{ij} \right|. \tag{4.20}$$

From the definition of a distance-d coloring, it is immediate from (4.20) that the probing approximation is exact if f is a polynomial of degree at most d, as all entries $[f(A)]_{ij}$ on the right hand side of (4.20) are zero in that case. More generally, if there is a strong exponential decay in $f(A)$ away from the sparsity pattern of A, one can expect the trace estimate to be rather accurate, as the terms in the sum will be small (in particular for large values of d). In [S8], which is discussed in Section 4.2.1, we perform a rigorous and in-depth analysis of this methodology.

4.2.1 A. Frommer, C. Schimmel, and M. Schweitzer, *Analysis of probing techniques for sparse approximation and trace estimation of decaying matrix functions*, SIAM J. Matrix Anal. Appl., 42 (2021)

Our paper [S8] discusses various important aspects of the probing methods that were introduced above, the most fundamental one probably being a detailed theoretical analysis of its error, which was lacking in the literature so far. Starting from (4.20), we derive several error estimates for the probing approximation (4.19), depending on the specific nature of the graph of A. For example, specialized results can be derived when A is banded or when G_A is a regular grid.

We present here only two of those results, one that is specific to regular grids and one that is applicable for general A.

When the graph of A is a regular D-dimensional grid, we provide an easily computable distance-d coloring in [S8, Theorem 2.2]. We then prove in [S8, Theorem 4.2] that if this coloring is used and the entries of $f(A)$ exhibit an exponential decay $|[f(A)]_{ij}| \leq Cq^{\mathrm{dist}(i,j)}$, the probing error satisfies

$$|\,\mathrm{tr}(f(A)) - \mathcal{T}_m(f(A))| \leq 2CDn\,\mathrm{Li}_{1-D}(q^d),$$

where $\mathrm{Li}_s(z) = \sum_{i=1}^{\infty} \frac{z^i}{i^s}$ is the polylogarithm.[52]

[52] Note that polylogarithms of negative integer order are rational functions of the form $\mathrm{Li}_{-s}(z) = \frac{p_s(z)}{(1-z)^{s+1}}$ where p_s is a polynomial of degree s with $p_s(0) = 0$.

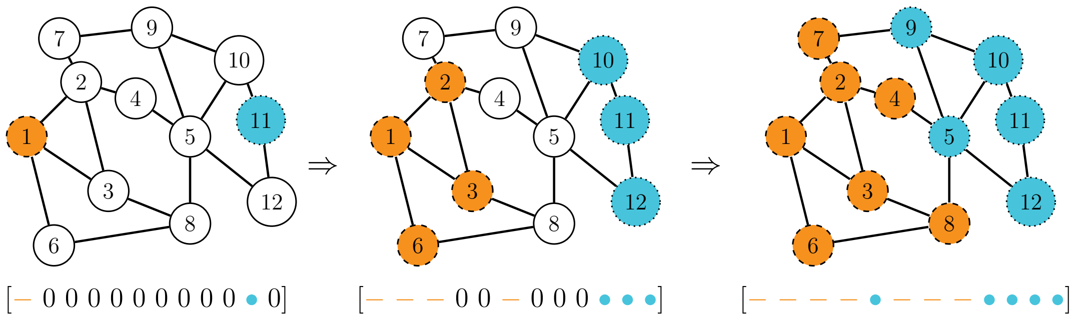


Figure 4.7: Spreading of the nonzero entries in the first three Arnoldi basis vectors, starting with $\boldsymbol{v}_\ell = \boldsymbol{e}_1 + \boldsymbol{e}_{11}$. Entries to which only the iteration corresponding to node 1 contributed are marked by an orange dash (–) and the corresponding nodes of the graph are marked with orange color and a dashed line. Entries to which only the iteration corresponding to node 11 contributed are marked by a blue dot (•) and the corresponding nodes of the graph are marked with blue color and a dotted line. Nodes corresponding to zero entries are filled white and have a solid line. As the nodes have a distance of 5, no mixing occurs in the first three basis vectors. *Originally [S8, Figure 5.1].*

A result for general A can be derived whenever a polynomial approximation property of the form

$$\min_{p_d \in \Pi_d} \max_{z \in W(A)} |f(z) - p_d(z)| \leq Cq^d \tag{4.21}$$

with $C > 0$, $0 < q < 1$ is available.[53] Assuming that (4.21) holds, we prove in [S8, Theorem 4.4] that

$$|\operatorname{tr}(f(A)) - \mathcal{T}_m(f(A))| \leq 2KCnq^d, \tag{4.22}$$

where $K = 1$ if A is normal and $K = 1 + \sqrt{2}$ otherwise. Note that the factor n on the right-hand side of (4.22) cannot be avoided and is actually sharp, as it is an absolute error bound; see also the numerical experiments reported in [S8, Section 6] which confirm this.

For computing the estimate (4.19), one needs to approximate $\boldsymbol{v}_\ell^H f(A)\boldsymbol{v}_\ell$, typically by a Krylov subspace method. This gives rise to the question how accurate these Krylov approximations need to be in order to not spoil the accuracy of the overall estimator, or, phrased differently, how many Krylov iterations one needs to perform.

To answer this, note that when starting the Arnoldi iteration with a vector of the form (4.18), the nonzeros in the basis vectors spread out "along the edges of G_A". As the probing vectors originate from a distance-d coloring, there is no "mixing" between the contributions of the different nodes in V_ℓ in the first $\lceil \frac{d+1}{2} \rceil$ iterations; see Figure 4.7 for an illustration.

[53] Actually, a similar result can also be obtained for any other form of polynomial approximation property, but for the ease of exposition we restrict ourselves to this specific type of bound.

Using this observation together with convergence bounds for Krylov methods, we balance the error of the trace estimate against the Krylov error and the error due to mixing in the basis vectors in order to determine the optimal number of Krylov steps to perform for each quadratic form. It turns out (and this is confirmed by numerical experiments) that if A is Hermitian, performing more than $\lceil \frac{d+1}{2} \rceil$ iterations cannot be expected to increase the overall accuracy, while for non-Hermitian A, at most d iterations should be performed; see [S8, Section 5] for details and [S8, Section 6.1] for a corresponding numerical experiment.

As additional contributions of the paper [S8], we present several other methods for efficiently determining a distance-d coloring, in particular a general purpose greedy method that is applicable without relying on a specific structure in A. We also investigate how to use probing to compute a sparse approximation of the whole matrix $f(A)$, which is, e.g., relevant for $f(z) = z^{-1}$ in the context of sparse approximate inverse preconditioners [26] or in inverse covariance estimation [36]).

We briefly note that in the recent preprint [84], we generalize and extend the results obtained in [S8] to the case of *stochastic probing*, where the nonzero entries of the probing vectors (4.18) are chosen as Rademacher random variables. In particular, we can improve over (4.22), showing that the error in the resulting method can in many cases be expected to scale with $\sqrt{n}$ rather than with n.

CHAPTER 5

CONTRIBUTIONS TO EACH PUBLICATION BY THE AUTHOR

Many of the publications covered in this cumulative habilitation thesis are the result of collaborative work. Therefore, this chapter contains a summary of the main contributions that I have made to each of these papers.

Restarted and sketched Krylov methods for matrix functions

[S5] **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.**

- Joint work on how to obtain suitable quadrature rules
- Working out details of the connection to restarting for Cauchy–Stieltjes functions as a special case
- Extension to Bernstein functions
- Joint implementation and experiments

[S9] **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.**

- General idea for comparison methodology

- Joint derivation of results for inexact (polynomial) inner solves
- Joint implementation and experiments

[S4] **M. A. Botchev, L. A. Knizhnerman, and M. Schweitzer, *Krylov subspace residual and restarting for certain second order differential equations*, SIAM J. Sci. Comput., 46 (2024), pp. S223–S253.**

- Idea to extend the residual time restarting framework to second-order ODEs
- Derivation of main formulas and proof of basic residual bounds
- Joint development of algorithmic details
- Joint implementation and experiments

[S10] **S. Güttel and M. Schweitzer, *Randomized sketching for Krylov approximations of large-scale matrix functions*, SIAM J. Matrix Anal. Appl., 44 (2023), pp. 1073–1095.**

- Idea to combine Krylov methods for $f(A)\boldsymbol{b}$ with randomized sketching
- Closed-form representation for sFOM approximant
- Joint work on convergence theory
- Algorithmic details (adaptive quadrature, stopping criterion)
- Joint implementation and experiments

Fréchet derivatives and low-rank updates of matrix functions

[S11] **P. Kandolf, A. Koskela, S. D. Relton, and M. Schweitzer, *Computing low-rank approximations of the Fréchet derivative of a matrix function using Krylov subspace methods*, Numer. Linear Algebra Appl., 28 (2021), p. e2401.**

- Idea to develop a (tensorized) Krylov method for approximating the whole Fréchet derivative instead of just its action on a vector
- Algorithmic details for the different considered methods
- Convergence theory for Cauchy–Stieltjes functions and the matrix logarithm
- Joint implementation and experiments

[S15] **M. Schweitzer, *Sensitivity of matrix function based network communicability measures: Computational methods and a priori bounds*, SIAM J. Matrix Anal. Appl., 44 (2023), pp. 1321–1348.**

- Single-authored

[S14] **M. Schweitzer, *Integral representations for higher-order Fréchet derivatives of matrix functions: Quadrature algorithms and new results on the level-2 condition number*, Linear Algebra Appl., 656 (2023), pp. 247–276.**

- Single-authored

[S1] **B. Arslan, S. D. Relton, and M. Schweitzer, *Structured level-2 condition numbers of matrix functions*, Electron. J. Linear Algebra, 40 (2024), pp. 28–47.**

- Development of approach for (quasi-)triangular matrices
- Proof of explicit formulas for the level-2 condition number in special cases
- Joint implementation and experiments

[S12] **K. Lund and M. Schweitzer, *The Fréchet derivative of the tensor t-function*, Calcolo, 60 (2023), p. 35.**

- Overall idea to investigate properties of the Fréchet derivative of the tensor t-function
- Proof of basic properties
- Idea of using the “shift operator” as tool in derivation of several results
- Theory and algorithms for t-function condition number and tensor nuclear norm
- Joint implementation and experiments

[S3] **B. Beckermann, D. Kressner, and M. Schweitzer, *Low-rank updates of matrix functions*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 539–565.**

- Development of algorithm (in particular for the non-Hermitian case)
- Proof of exactness property for polynomials
- Idea of using the integral representation of Cauchy–Stieltjes functions for the convergence analysis
- Implementation and experiments

[S2] **B. Beckermann, A. Cortinovis, D. Kressner, and M. Schweitzer, *Low-rank updates of matrix functions II: Rational Krylov methods*, SIAM J. Numer. Anal., 59 (2021), pp. 1325–1347.**

- Proof of exactness property for rational functions
- Investigation of relation to approach from [31]
- Development of algorithm and theory for updates of the matrix sign function
- Implementation and experiments

Decay bounds and probing methods

[S6] **A. Frommer, C. Schimmel, and M. Schweitzer, *Bounds for the decay of the entries in inverses and Cauchy–Stieltjes functions of certain sparse, normal matrices*, Numer. Linear Algebra Appl., 25 (2018), p. e2131**

- Proof of results for Cauchy–Stieltjes functions
- Comparison to [58] and [79]
- Joint implementation and experiments

[S7] **A. Frommer, C. Schimmel, and M. Schweitzer, *Non-Toeplitz decay bounds for inverses of Hermitian positive definite tridiagonal matrices*, Electron. Trans. Numer. Anal., 48 (2018), pp. 362–372.**

- Idea for finding decay bounds that do not have a Toeplitz structure
- Development of block-partitioning methodology for finding decay bounds
- Joint implementation and experiments

[S13] **M. Schweitzer, *Decay bounds for Bernstein functions of Hermitian matrices with applications to the fractional graph Laplacian*, Electron. Trans. Numer. Anal., 55 (2022), pp. 438–454.**

- Single-authored

[S8] **A. Frommer, C. Schimmel, and M. Schweitzer, *Analysis of probing techniques for sparse approximation and trace estimation of decaying matrix functions*, SIAM J. Matrix Anal. Appl., 42 (2021), pp. 1290–1318.**

- Proof of error bounds for D-dimensional lattices involving the polylogarithm
- Investigation of implications for Krylov subspace methods
- Joint implementation and experiments

BIBLIOGRAPHY

[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.

[2] A. H. Al-Mohy, *Conditioning of matrix functions of quasi-triangular matrices*, SIAM J. Matrix Anal. Appl., 45 (2024), pp. 954–966.

[3] A. H. Al-Mohy and B. Arslan, *The complex step approximation to the higher order Fréchet derivatives of a matrix function*, Numer. Algorithms, 87 (2021), pp. 1061–1074.

[4] H. Alzer and C. Berg, *Some classes of completely monotonic functions*, Ann. Acad. Sci. Fenn., Math., 27 (2002), pp. 445–460.

[5] S. Amat, S. Busquier, and J. Gutiérrez, *Geometric constructions of iterative functions to solve nonlinear equations*, J. Comput. Appl. Math., 157 (2003), pp. 197–205.

[6] W. E. Arnoldi, *The principle of minimized iteration in the solution of the matrix eigenvalue problem*, Q. Appl. Math., 9 (1951), pp. 17–29.

[7] B. Arslan, V. Noferini, and F. Tisseur, *The structured condition number of a differentiable map between matrix manifolds, with applications*, SIAM J. Matrix Anal. Appl., 40 (2019), pp. 774–799.

[8] E. Aune, D. P. Simpson, and J. Eidsvik, *Parameter estimation in high dimensional Gaussian distributions*, Stat. Comput., 24 (2014), pp. 247–263.

[9] H. Avron, P. Maymounkov, and S. Toledo, *Blendenpik: Supercharging LAPACK's least-squares solver*, SIAM J. Sci. Comput., 32 (2010), pp. 1217–1236.

[10] O. Balabanov and L. Grigori, *Randomized block Gram-Schmidt process for solution of linear systems and eigenvalue problems*, arXiv preprint arXiv:2111.14641, (2021).

[11] ——, *Randomized Gram–Schmidt process with application to GMRES*, SIAM J. Sci. Comput., 44 (2022), pp. A1450–A1474.

[12] O. Balabanov and A. Nouy, *Randomized linear algebra for model reduction. Part I: Galerkin methods and error estimation*, Adv. Comput. Math., 45 (2019), pp. 2969–3019.

[13] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*, vol. 15, Springer Science & Business Media, 2003.

[14] A. H. Bentbib, M. El Ghomari, K. Jbilou, and L. Reichel, *The global Golub-Kahan method and Gauss quadrature for tensor function approximation*, Numer. Algorithms, 92 (2023), pp. 5–34.

[15] M. Benzi, *Localization in Matrix Computations: Theory and Applications*, in Exploiting Hidden Structure in Matrix Computations: Algorithms and Applications, M. Benzi and V. Simoncini, eds., vol. 2173 of C.I.M.E. Foundation Subseries, Springer, New York, 2016, pp. 211–317.

[16] M. Benzi, D. Bertaccini, F. Durastante, and I. Simunec, *Nonlocal network dynamics via fractional graph Laplacians*, J. Complex Netw., 8 (2020), p. cnaa017.

[17] M. Benzi and P. Boito, *Matrix functions in network analysis*, GAMM-Mitteilungen, 43 (2020), p. e202000012.

[18] M. Benzi and G. H. Golub, *Bounds for the entries of matrix functions with applications to preconditioning*, BIT, 39 (1999), pp. 417–438.

[19] M. Benzi and C. Klymko, *Total communicability as a centrality measure*, J. Complex Netw., 1 (2013), pp. 124–149.

[20] M. Benzi and N. Razouk, *Decay bounds and $O(n)$ algorithms for approximating functions of sparse matrices*, Electron. Trans. Numer. Anal., 28 (2007), pp. 16–39.

[21] M. Benzi and M. Rinelli, *Refined decay bounds on the entries of spectral projectors associated with sparse Hermitian matrices*, Linear Algebra Appl., 647 (2022), pp. 1–30.

[22] M. Benzi, M. Rinelli, and I. Simunec, *Computation of the von Neumann entropy of large matrices via trace estimators and rational Krylov methods*, Numer. Math., 155 (2023), pp. 377–414.

[23] M. Benzi and V. Simoncini, *Decay bounds for functions of Hermitian matrices with banded or Kronecker structure*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 1263–1282.

[24] M. Benzi and V. Simoncini, *Approximation of functions of large matrices with Kronecker structure*, Numer. Math., 135 (2017), pp. 1–26.

[25] M. Benzi and I. Simunec, *Rational Krylov methods for fractional diffusion problems on graphs*, BIT, (2021), pp. 1–29.

[26] M. Benzi and M. Tuma, *A comparative study of sparse approximate inverse preconditioners*, Appl. Numer. Math., 30 (1999), pp. 305–340.

[27] C. Berg, *Stieltjes-Pick-Bernstein-Schoenberg and their connection to complete monotonicity*, in Positive Definite Functions. From Schoenberg to Space-Time Challenges, J. Mateu and E. Porcu, eds., Dept. of Mathematics, University Jaume I, Castellón de la Plana, Spain, 2008.

[28] C. Berg and G. Forst, *Potential Theory on Locally Compact Abelian Groups*, Springer, Berlin Heidelberg, 1975.

[29] M. Berljafa, S. Elsworth, and S. Güttel, *A rational Krylov toolbox for MATLAB*, tech. rep., Manchester Institute for Mathematical Sciences, The University of Manchester, 2014. MIMS EPrint 2014.56.

[30] M. Berljafa and S. Güttel, *Generalized rational Krylov decompositions with an application to rational approximation*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 894–916.

[31] D. S. Bernstein and Ch. F. van Loan, *Rational matrix functions and rank-1 updates*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 145–154.

[32] D. Bertaccini and F. Durastante, *Nonlocal diffusion of variable order on complex networks*, Int. J. Comput. Math., 7 (2022), pp. 172–191.

[33] D. Bianchi, M. Donatelli, F. Durastante, and M. Mazza, *Compatibility, embedding and regularization of non-local random walks on graphs*, J. Math. Anal. Appl., 511 (2022), p. 126020.

[34] D. Bini, S. Dendievel, G. Latouche, and B. Meini, *Computing the exponential of large block-triangular block-Toeplitz matrices encountered in fluid queues*, Linear Algebra Appl., 502 (2016), pp. 387–419.

[35] D. A. Bini, G. Latouche, and B. Meini, *Numerical Methods for Structured Markov Chains*, Oxford University Press, 2005.

[36] M. Bollhöfer, A. Eftekhari, S. Scheidegger, and O. Schenk, *Large-scale sparse inverse covariance matrix estimation*, SIAM J. Sci. Comput., 41 (2019), pp. A380–A401.

[37] A. Boriçi, *Fast methods for computing the Neuberger operator*, in Numerical Challenges in Lattice Quantum Chromodynamics, A. Frommer, T. Lippert, B. Medeke, and K. Schilling, eds., Berlin, Heidelberg, 2000, Springer Berlin Heidelberg, pp. 40–47.

[38] M. A. Botchev, V. Grimm, and M. Hochbruck, *Residual, restarting, and Richardson iteration for the matrix exponential*, SIAM J. Sci. Comput., 35 (2013), pp. A1376–A1397.

[39] M. A. Botchev, D. Harutyunyan, and J. J. W. van der Vegt, *The Gautschi time stepping scheme for edge finite element discretizations of the Maxwell equations*, J. Comput. Phys., 216 (2006), pp. 654–686.

[40] M. A. Botchev and L. A. Knizhnerman, *ART: Adaptive residual-time restarting for Krylov subspace matrix exponential evaluations*, J. Comput. Appl. Math., 364 (2020), p. 112311.

[41] M. A. Botchev, L. A. Knizhnerman, and E. E. Tyrtyshnikov, *Residual and restarting in Krylov subspace evaluation of the φ function*, SIAM J. Sci. Comput., 43 (2021), pp. A3733–A3759.

[42] A. Bouras and V. Frayssé, *Inexact matrix-vector products in Krylov methods for solving linear systems: A relaxation strategy*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 660–678.

[43] D. R. Bowler and T. Miyazaki, *$O(N)$ methods in electronic structure calculations*, Rep. Prog. Phys., 75 (2012), p. 036503.

[44] K. Braman, *Third-order tensors as linear operators on a space of matrices*, Linear Algebra Appl., 433 (2010), pp. 1241–1253.

[45] L. Burke and S. Güttel, *Krylov subspace recycling with randomized sketching for matrix functions*, arXiv preprint arXiv:2308.02290, (2023).

[46] E. Celledoni and I. Moret, *A Krylov projection method for systems of ODEs*, Appl. Numer. Math., 24 (1997), pp. 365–378.

[47] T. Chen, A. Greenbaum, C. Musco, and C. Musco, *Error bounds for Lanczos-based matrix function approximation*, SIAM J. Matrix Anal. Appl., 43 (2022), pp. 787–811.

[48] A. Cortinovis, D. Kressner, and Y. Nakatsukasa, *Speeding up Krylov subspace methods for computing $f(A)b$ via randomization*, SIAM J. Matrix Anal. Appl., 45 (2024), pp. 619–633.

[49] M. Crouzeix, *Numerical range and functional calculus in Hilbert space*, J. Funct. Anal., 244 (2007), pp. 668–690.

[50] M. Crouzeix and D. Kressner, *A bivariate extension of the Crouzeix-Palencia result with an application to Fréchet derivatives of matrix functions*, arXiv preprint arXiv:2007.09784, (2020).

[51] 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.

[52] P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration*, Academic Press, New York, 1975.

[53] C. de Boor, *Divided differences*, Surv. Approx. Theory, 1 (2005), pp. 46–69.

[54] M. De Domenico, *Multilayer network dataset.* https://manliodedomenico.com/data.php.

[55] M. De Domenico, A. Solé-Ribalta, S. Gómez, and A. Arenas, *Navigability of interconnected networks under random failures*, Proc. Natl. Acad. Sci., 111 (2014), pp. 8351–8356.

[56] O. De la Cruz Cabrera, J. Jin, S. Noschese, and L. Reichel, *Communication in complex networks*, Appl. Numer. Math., 172 (2022), pp. 186–205.

[57] S. Demko, *Inverses of band matrices and local convergence of spline projections*, SIAM J. Numer. Anal., 14 (1977), pp. 616–619.

[58] S. Demko, W. F. Moss, and W. Smith, *Decay rates for inverses of banded matrices*, Math. Comp., 43 (1984), pp. 491–499.

[59] P. A. M. Dirac, *The Principles of Quantum Mechanics, 4th edition*, Oxford University Press, Oxford, 1958.

[60] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, *Subspace sampling and relative-error matrix approximation: Column-based methods*, in Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, Springer, 2006, pp. 316–326.

[61] 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.

[62] V. Druskin and L. Knizhnerman, *Two polynomial methods of calculating functions of symmetric matrices*, U.S.S.R. Comput. Math. Math. Phys., 29 (1989), pp. 112–121.

[63] 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.

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

[65] 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.

[66] M. Eiermann, O. G. Ernst, and S. Güttel, *Deflated restarting for matrix functions*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 621–641.

[67] V. Eijkhout and B. Polman, *Decay rates of inverses of banded M-matrices that are near to Toeplitz matrices*, Linear Algebra Appl., 109 (1988), pp. 247–277.

[68] J. van den Eshof, A. Frommer, Th. 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.

[69] J. van den Eshof and M. Hochbruck, *Preconditioning Lanczos approximations to the matrix exponential*, SIAM J. Sci. Comput., 27 (2006), pp. 1438–1457.

[70] J. Van Den Eshof and G. L. Sleijpen, *Inexact Krylov subspace methods for linear systems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 125–153.

[71] E. Estrada, *The Structure of Complex Networks: Theory and Applications*, Oxford University Press, 2012.

[72] ——, *Path Laplacians versus fractional Laplacians as nonlocal operators on networks*, New J. Phys., 23 (2021), p. 073049.

[73] E. Estrada and N. Hatano, *Communicability in complex networks*, Phys. Rev. E, 77 (2008), p. 036111.

[74] E. Estrada, D. J. Higham, and N. Hatano, *Communicability betweeness in complex networks*, Physica A, 388 (2009), pp. 764–774.

[75] E. Estrada and J. A. Rodriguez-Velazquez, *Subgraph centrality in complex networks*, Phys. Rev. E, 71 (2005), p. 056103.

[76] M. Fasi, N. J. Higham, and X. Liu, *Computing the square root of a low-rank perturbation of the scaled identity matrix*, SIAM J. Matrix Anal. Appl., 44 (2023), pp. 156–174.

[77] G. Fertin, E. Godard, and A. Raspaud, *Acyclic and k-distance coloring of the grid*, Inf. Process. Lett., 87 (2003), pp. 51–58.

[78] N. J. Ford, D. V. Savostyanov, and N. L. Zamarashkin, *On the decay of the elements of inverse triangular Toeplitz matrices*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 1288–1302.

[79] R. Freund, *On polynomial approximations to $f_a(z) = (z - a)^{-1}$ with complex a and some applications to certain non-Hermitian matrices*, Approx. Theory its Appl., 5 (1989), pp. 15–31.

[80] 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.

[81] 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.

[82] A. Frommer, K. Kahl, S. Krieg, B. Leder, and M. Rottmann, *Adaptive aggregation-based domain decomposition multigrid for the lattice Wilson–Dirac operator*, SIAM J. Sci. Comput., 36 (2014), pp. A1581–A1608.

[83] A. Frommer and P. Maass, *Fast CG-based methods for Tikhonov–Phillips regularization*, SIAM J. Sci. Comput., 20 (1999), pp. 1831–1850.

[84] A. Frommer, M. Rinelli, and M. Schweitzer, *Analysis of stochastic probing methods for estimating the trace of functions of sparse symmetric matrices*, arXiv preprint arXiv:2308.07722, (2023).

[85] A. Frommer and V. Simoncini, *Matrix functions*, in Model Order Reduction: Theory, Research Aspects and Applications, W. H. A. Schilders, H. A. van der Vorst, and J. Rommes, eds., Springer, Berlin Heidelberg, 2008, pp. 275–303.

[86] W. Gautschi, *Numerical integration of ordinary differential equations based on trigonometric polynomials*, Numer. Math., 3 (1961), pp. 381–397.

[87] P. H. Ginsparg and K. G. Wilson, *A remnant of chiral symmetry on the lattice*, Phys. Rev. D, 25 (1982), pp. 2649–2657.

[88] P.-L. Giscard, K. Lui, S. Thwaite, and D. Jaksch, *An exact formulation of the time-ordered exponential using path-sums*, J. Math. Phys., 56 (2015), p. 053503.

[89] S. Güttel, *Rational Krylov Methods for Operator Functions*, PhD thesis, Fakultät für Mathematik und Informatik der Technischen Universität Bergakademie Freiberg, 2010.

[90] S. Güttel, *Rational Krylov approximation of matrix functions: Numerical methods and optimal pole selection*, GAMM-Mitt., 36 (2013), pp. 8–31.

[91] S. Güttel and L. Knizhnerman, *A black-box rational Arnoldi variant for Cauchy–Stieltjes matrix functions*, BIT, 53 (2013), pp. 595–616.

[92] N. Halko, P.-G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288.

[93] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Natl. Bur. Stand., 49 (1952), pp. 409–436.

[94] N. J. Higham, *Functions of Matrices: Theory and Computation*, SIAM, Philadelphia, 2008.

[95] N. J. Higham and S. D. Relton, *Higher order Fréchet derivatives of matrix functions and the level-2 condition number*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 1019–1037.

[96] ——, *Estimating the largest elements of a matrix*, SIAM J. Sci. Comput., 38 (2016), pp. C584–C601.

[97] E. Hille and R. S. Phillips, *Functional analysis and semi-groups*, vol. 31, American Mathematical Society, 1996.

[98] M. Hochbruck and M. E. Hochstenbach, *Subspace extraction for matrix functions*, tech. rep., Case Western Reserve University, Department of Mathematics, Cleveland, 2005.

[99] M. Hochbruck and Ch. Lubich, *On Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., 34 (1997), pp. 1911–1925.

[100] M. Hochbruck and Ch. Lubich, *A Gautschi-type method for oscillatory second-order differential equations*, Numer. Math., 83 (1999), pp. 403–426.

[101] M. Hochbruck, Ch. Lubich, and H. Selhofer, *Exponential integrators for large systems of differential equations*, SIAM J. Sci. Comput., 19 (1998), pp. 1552–1574.

[102] M. Hochbruck and A. Ostermann, *Exponential integrators*, Acta Numerica, 19 (2010), pp. 209–286.

[103] M. F. Hutchinson, *A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines*, Commun. Stat. Simul., 19 (1990), pp. 433–450.

[104] B. Iannazzo, B. Jeuris, and F. Pompili, *The derivative of the matrix geometric mean with an application to the nonnegative decomposition of tensor grids*, in Structured Matrices in Numerical Linear Algebra: Analysis,

Algorithms and Applications, D. A. Bini, F. Di Benedetto, E. Tyrtyshnikov, and M. Van Barel, eds., Springer International Publishing, 2019, pp. 107–128.

[105] 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.

[106] A. Iserles, *How large is the exponential of a banded matrix?*, N. Z. J. Math., 29 (2000), pp. 177–192.

[107] C. Jagels and L. Reichel, *The extended Krylov subspace method and orthogonal Laurent polynomials*, Linear Algebra Appl., 431 (2009), pp. 441–458.

[108] ——, *Recursion relations for the extended Krylov subspace method*, Linear Algebra Appl., 434 (2011), pp. 1716–1732.

[109] D. P. Jagger, *MATLAB toolbox for classical matrix groups*, M.Sc. Thesis, University of Manchester, Manchester, England, 2003.

[110] D. Kershaw, *Inequalities on the elements of the inverse of a certain tridiagonal matrix*, Math. Comput., (1970), pp. 155–158.

[111] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, *Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 148–172.

[112] M. E. Kilmer and C. D. Martin, *Factorization strategies for third-order tensors*, Linear Algebra Appl., 435 (2011), pp. 641–658.

[113] L. Knizhnerman, *Calculation of functions of unsymmetric matrices using Arnoldi's method*, Zh. Vychisl. Mat. Mat. Fiz., 31 (1991), pp. 1–9.

[114] 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.

[115] D. Kressner, *Bivariate matrix functions*, Oper. Matrices, 8 (2014), pp. 449–466.

[116] ——, *A Krylov subspace method for the approximation of bivariate matrix functions*, in Structured Matrices in Numerical Linear Algebra: Analysis, Algorithms and Applications, D. A. Bini, F. Di Benedetto, E. Tyrtyshnikov, and M. Van Barel, eds., Springer, 2019, pp. 197–214.

[117] D. Kressner and A. Šušnjara, *Fast computation of spectral projectors of banded matrices*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 984–1009.

[118] J. Laeuchli and A. Stathopoulos, *Extending hierarchical probing for computing the trace of matrix inverses*, SIAM J. Sci. Comput., 42 (2020), pp. A1459–A1485.

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

[120] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Natl. Stand., 45 (1950), pp. 255–282.

[121] J. Liesen and Z. Strakoš, *Krylov Subspace Methods: Principles and Analysis*, Oxford University Press, 2013.

[122] L. Lopez and A. Pugliese, *Decay behaviour of functions of skew-symmetric matrices*, Proceedings of HERCMA, (2005), pp. 22–24.

[123] K. Lund, *The tensor t-function: A definition for functions of third-order tensors*, Numer. Linear Algebra Appl., 27 (2020), p. e2288.

[124] J. R. Martins, P. Sturdza, and J. J. Alonso, *The complex-step derivative approximation*, ACM Trans. Math. Softw., 29 (2003), pp. 245–262.

[125] P.-G. Martinsson and J. A. Tropp, *Randomized numerical linear algebra: Foundations and algorithms*, Acta Numerica, 29 (2020), pp. 403–572.

[126] S. Massei and L. Robol, *Rational Krylov for Stieltjes matrix functions: Convergence and pole selection*, BIT, (2021), pp. 237–273.

[127] R. Mathias, *A chain rule for matrix functions and applications*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 610–620.

[128] G. Meinardus, *Approximation of Functions: Theory and Numerical Methods*, vol. 13, Springer Science & Business Media, 2012.

[129] Met Office, *Cartopy: A cartographic Python library with a Matplotlib interface*, Exeter, Devon, 2010 - 2015.

[130] G. Meurant, *A review on the inverse of symmetric tridiagonal and block tridiagonal matrices*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 707–728.

[131] R. A. Meyer, C. Musco, C. Musco, and D. P. Woodruff, *Hutch++: Optimal stochastic trace estimation*, in Symposium on Simplicity in Algorithms (SOSA), SIAM, 2021, pp. 142–155.

[132] I. Moret and P. Novati, *RD-rational approximations of the matrix exponential.*, BIT, 44 (2004), pp. 595–615.

[133] Y. Nakatsukasa and J. A. Tropp, *Fast & accurate randomized algorithms for linear systems and eigenvalue problems*, arXiv preprint arXiv:2111.00113, (2021).

[134] H. Neuberger, *Exactly massless quarks on the lattice*, Phys. Lett., B, 417 (1998), pp. 141–144.

[135] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.

[136] D. Palitta, S. Pozza, and V. Simoncini, *The short-term rational Lanczos method and applications*, SIAM J. Sci. Comput., 44 (2022), pp. A2843–A2870.

[137] D. Palitta, M. Schweitzer, and V. Simoncini, *Sketched and truncated polynomial Krylov methods: Evaluation of matrix functions*, tech. rep., 2023. arXiv: 2306.06481.

[138] T. Park and Y. Nakatsukasa, *Approximating sparse matrices and their functions using matrix-vector products*, arXiv:2310.05625 [math.NA], Oct. 2023.

[139] S. Pozza and V. Simoncini, *Inexact Arnoldi residual estimates and decay properties for functions of non-Hermitian matrices*, BIT, 59 (2019), pp. 969–986.

[140] S. Pozza and F. Tudisco, *On the stability of network indices defined by means of matrix functions*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 1521–1546.

[141] E. H. Rubensson, *A unifying framework for higher order derivatives of matrix functions*, SIAM J. Matrix Anal. Appl., 45 (2024), pp. 504–528.

[142] A. Ruhe, *Rational Krylov sequence methods for eigenvalue computation*, Linear Algebra Appl., 58 (1984), pp. 391–405.

[143] ——, *Rational Krylov algorithms for nonsymmetric eigenvalue problems*, IMA Vol. Math. Appl., 60 (1994), pp. 149–164.

[144] Y. Saad, *Analysis of some Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., 29 (1992), pp. 209–228.

[145] Y. Saad and M. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.

[146] T. Sarlos, *Improved approximation algorithms for large matrices via random projections*, in 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06), IEEE, 2006, pp. 143–152.

[147] R. L. Schilling, R. Song, and Z. Vondracek, *Bernstein Functions – Theory and Applications*, De Gruyter, Berlin, Boston, 2012.

[148] C. Schimmel, *Bounds for the Decay in Matrix Functions and its Exploitation in Matrix Computations*, PhD thesis, Bergische Universität Wuppertal, 2019.

[149] M. Schweitzer, *Monotone convergence of the extended Krylov subspace method for Laplace–-Stieltjes functions of Hermitian positive definite matrices*, Linear Algebra Appl., 507 (2016), pp. 486–498.

[150] A. Sharp, *Distance coloring*, in European Symposium on Algorithms, Springer, 2007, pp. 510–521.

[151] J. Sherman and W. J. Morrison, *Adjustment of an inverse matrix corresponding to a change in one element of a given matrix*, Ann. Math. Stat., 21 (1950), pp. 124–127.

[152] S. Shumeli, P. Drineas, and H. Avron, *Low-rank updates of matrix square roots*, Numer. Linear Algebra Appl., 31 (2024), p. e2528.

[153] V. Simoncini, *A new iterative method for solving large-scale Lyapunov matrix equations*, SIAM J. Sci. Comput., 29 (2007), pp. 1268–1288.

[154] ——, *Extended Krylov subspace for parameter dependent systems*, Appl. Numer. Math., 60 (2010), pp. 550–560.

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

[156] V. Simoncini and D. B. Szyld, *Theory of inexact Krylov subspace methods and applications to scientific computing*, SIAM J. Sci. Comput., 25 (2003), pp. 454–477.

[157] C. Sohler and D. P. Woodruff, *Subspace embeddings for the L1-norm with applications*, in Proceedings of the forty-third annual ACM symposium on theory of computing, 2011, pp. 755–764.

[158] A. Stathopoulos, J. Laeuchli, and K. Orginos, *Hierarchical probing for estimating the trace of the matrix inverse on toroidal lattices*, SIAM J. Sci. Comput., 35 (2013), pp. 299–322.

[159] H. M. Switzer, A. Stathopoulos, E. Romero, J. Laeuchli, and K. Orginos, *Probing for the trace estimation of a permuted matrix inverse corresponding to a lattice displacement*, SIAM J. Sci. Comput., 44 (2022), pp. B1096–B1121.

[160] 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.

[161] J. M. Tang and Y. Saad, *A probing method for computing the diagonal of a matrix inverse*, Numer. Linear Algebra Appl., 19 (2012), pp. 485–501.

[162] D. Thanou, X. Dong, D. Kressner, and P. Frossard, *Learning heat diffusion graphs*, IEEE Trans. Signal Inform. Process. Netw., 3 (2017), pp. 484–499.

[163] E. Timsit, L. Grigori, and O. Balabanov, *Randomized orthogonal projection methods for Krylov subspace solvers*, arXiv preprint arXiv:2302.07466, (2023).

[164] J. A. Tropp, *Improved analysis of the subsampled randomized Hadamard transform*, Adv. Adapt. Data Anal., 3 (2011), pp. 115–126.

[165] M. Tsolakis, *Efficient Computation of the Action of Matrix Rational Functions and Laplace Transforms*, Ph.D. thesis, Bergische Universität Wuppertal, 2023.

[166] 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.

[167] T. Werner, *On using the complex step method for the approximation of Fréchet derivatives of matrix functions in automorphism groups*, GAMMAS, 4 (2022), pp. 49–64.

[168] K. G. Wilson, *Quarks and strings on a lattice*, in New Phenomena in Subnuclear Physics. Part A., A. Zichichi, ed., Plenum Press, New York, 1977, pp. 69–125.

[169] M. A. Woodbury, *Inverting modified matrices*, Department of Statistics, Princeton University, 1950.

[170] D. P. Woodruff, *Sketching as a tool for numerical linear algebra*, Found. Trends Theor. Comput. Sci., 10 (2014), pp. 1–157.

[171] F. Woolfe, E. Liberty, V. Rokhlin, and M. Tygert, *A fast randomized algorithm for the approximation of matrices*, Appl. Comput. Harmon. Anal., 25 (2008), pp. 335–366.

[172] G. Zolotarev, *Application of elliptic functions to the problem of functions which vary the least or the most from zero*, Abh. St. Petersb., 30 (1877), pp. 1–59.