KRYLOV SUBSPACE RESTARTING FOR MATRIX LAPLACE TRANSFORMS

A. FROMMER*, K. KAHL*, M. SCHWEITZER*, AND M. TSOLAKIS*

Abstract. A common way to approximate $F(A)b$—the action of a matrix function on a vector—is to use the Arnoldi approximation. Since a new vector needs to be generated and stored in every iteration, one is often forced to rely on restart algorithms which are either not efficient, not stable or only applicable to restricted classes of functions. We present a new representation of the error of the Arnoldi iterates if the function F is given as a Laplace transform. Based on this representation we build an efficient and stable restart algorithm. In doing so we extend earlier work for the class of Stieltjes functions which are special Laplace transforms. We report several numerical experiments including comparisons with the restart method for Stieltjes functions.

Key words. matrix functions, Krylov subspace methods, restarted Arnoldi method, Laplace transform, quadrature

AMS subject classifications. 65F60, 65F50, 44A10, 65D30, 65D07

MATLAB package available at https://github.com/MaTso7/laplace_restarting

1. Introduction. Computing $F(A)b$, the action of a matrix function $F(A) \in \mathbb{C}^{n \times n}$ on a vector $b \in \mathbb{C}^n$, is an important task in many scientific computing applications, including exponential integrators for differential equations [35], network analysis [8], theoretical particle physics [26], machine learning [40] and many others. In these applications, the matrix $A \in \mathbb{C}^{n \times n}$ is typically very large and sparse, so that explicitly forming $F(A)$—which is a dense matrix in general, irrespective of the sparsity of A—is not feasible with regard to both complexity and memory requirements. Therefore, one has to resort to *iterative methods* that directly approximate the vector $F(A)b$. The most widely used classes of algorithms for this task are polynomial [23, 42] and rational [24, 27, 32, 38] Krylov subspace methods.

While rational Krylov methods can greatly outperform polynomial methods when they are applicable (e.g., when shifted linear systems with A can be efficiently solved by a sparse direct solver), there are situations in which polynomial methods are superior: The size and sparsity pattern of A might make the direct solution of shifted linear systems infeasible, or A might only be implicitly available through a routine that returns the result of the matrix-vector product. In these cases, combining an "outer" rational Krylov method with an "inner" polynomial iterative solver for linear systems is in general not advisable (unless a *very* efficient preconditioner is available); see [33]. Thus—despite the tremendous theoretical and algorithmic advances in the area of rational Krylov methods in recent years—polynomial Krylov methods are still of utmost importance, in particular for large-scale applications.

However, for very large matrix sizes n, polynomial Krylov methods present challenges on their own: Their backbone is the Arnoldi process [6] (which reduces to the short-recurrence Lanczos method [37] when A is Hermitian), which computes a nested orthonormal basis $v_1, \ldots, v_m$ of the Krylov subspace

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

Collecting the basis vectors in $V_m = [v_1 \mid \cdots \mid v_m] \in \mathbb{C}^{n \times m}$ and the coefficients from the orthogonalization process in an upper Hessenberg matrix $H_m \in \mathbb{C}^{m \times m}$, one has

---
*School of Mathematics and Natural Sciences, Bergische Universität Wuppertal, 42097 Wuppertal, Germany, {frommer,kkahl,marcel,tsolakis}@uni-wuppertal.de

the *Arnoldi relation*

$$(1.1) \qquad AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T,$$

where $e_m \in \mathbb{C}^m$ denotes the mth canonical unit vector. Given (1.1), one obtains the *Arnoldi approximation* $f_m \in \mathcal{K}_m(A, b)$ for $F(A)b$ by projecting the original problem onto the Krylov space, i.e.,

$$(1.2) \qquad f_m := V_m F(V_m^H A V_m) V_m^H b = \|b\|_2 V_m F(H_m) e_1,$$

where $\|\cdot\|_2$ denotes the Euclidean norm and $e_1 \in \mathbb{C}^m$ is the first canonical unit vector. A large computational burden associated with forming the approximation (1.2) is that it requires storing all vectors of the orthonormal basis $v_1, \dots, v_m$. This can quickly surpass the available memory if n is very large. This is also the case when A is Hermitian, in contrast to the situation for linear systems, where the short recurrence for the basis vectors translates into a short recurrence for the iterates in, for example, the conjugate gradient method [34]. Thus, without appropriate countermeasures, the approximation accuracy that is reachable by a Krylov method might be limited by the available memory. Additionally, when A is non-Hermitian, the orthogonalization of basis vectors becomes more and more expensive with growing m, rendering Krylov methods much less efficient when a large number of iterations is required.

A remedy is *restarting*: After a fixed (small) number $m_{\max}$ of iterations, the Arnoldi approximation $f_{m_{\max}}$ is formed. Then, the matrices V_m and H_m are discarded, and a new Arnoldi process is started for approximating the remaining *error*

$$(1.3) \qquad \varepsilon_{m_{\max}} := F(A)b - f_{m_{\max}}.$$

Clearly, approximating (1.3) by the Arnoldi method requires that it can again be written as the action of a matrix function on a vector. While this is evidently the case when solving linear systems (i.e., for $F(s) = s^{-1}$), where the error fulfills the residual equation $A\varepsilon_{m_{\max}} = r_{m_{\max}}$ with $r_{m_{\max}} = b - Af_{m_{\max}}$, the situation is much more difficult for general functions F. Over the last fifteen years, numerous publications have investigated how to transfer the restarting approach to general matrix functions [1,2,25,29,36,45,48]. All of these approaches have in common that they are either only applicable to certain classes of functions (i.e., they are not fully general) or they have shortcomings with regard to numerical stability or computational efficiency (e.g., the required work grows from one restart cycle to the next). In this work, we add to the current state of the art in restarted Krylov methods by explaining—both theoretically and algorithmically—how restarts are possible when F results from the Laplace transform of some function f. This covers a large class of practically relevant functions and thus significantly extends the scope in which restarting is possible in a numerically efficient and stable way. While the approach based on the Cauchy integral formula presented in [29] is in theory applicable to any analytic function, and thus also to Laplace transforms, its practical implementation requires choosing a suitable contour Γ in the complex plane and a corresponding quadrature rule. When choosing these, one needs knowledge on the position of the spectrum of A in the complex plane and has to take specifics of the function F to be approximated into account as otherwise the quadrature rule on Γ might converge very slowly or become unstable. In contrast, the method we propose here works for general Laplace transforms, only employing simple, general purpose Gauss-Kronrod quadrature rules on the positive real axis. The only requirement is that the field of values lies within the region of absolute convergence of the Laplace transform, the typical case being that A is positive

definite (not necessarily Hermitian) in the sense that $\mathrm{Re}(x^H Ax) > 0$ for $x \neq 0$, and the Laplace transform converges absolutely in the right half plane.

The remainder of this paper is organized as follows. Important basic material is covered in section 2. In particular, we review the related Arnoldi restarting approach for Stieltjes matrix functions, and we collect some basic facts about the Laplace transform. In section 3, we lay the necessary theoretical foundations for restarting for matrix Laplace transforms and explain how it relates to other restart approaches. We explain in section 4 how the method can be extended to two related classes of functions. Section 5 is devoted to several implementation aspects that are crucial for making the method feasible in practice. In section 6, we illustrate the performance of our method and compare it to existing alternatives in a series of numerical experiments on both academic and real-world benchmark problems. Concluding remarks are given in section 7.

2. Basics. In this section, we review some basic material on which we build in later sections.

2.1. The restarted Arnoldi method for Stieltjes functions. We start by going into the details of the quadrature-based restarting approach for Stieltjes matrix functions from [29,45], as it is closely related to the present work; see also Corollary 3.5 below.

Given a maximum Krylov subspace size m (dictated, e.g., by the available memory), the basic idea of the restarted Arnoldi method for $F(A)b$ is to compute a sequence of (hopefully) more and more accurate approximations $f_m^{(1)}, f_m^{(2)}, f_m^{(3)}, \ldots$, where $f_m^{(1)} = f_m = \|b\|_2 V_m F(H_m)e_1$ is the usual m-step Arnoldi approximation (1.2) and further iterates (belonging to the kth *Arnoldi cycle*) are defined via the recurrence

$$(2.1) \qquad\qquad f_m^{(k+1)} = f_m^{(k)} + d_m^{(k)}, \quad k \geq 1$$

where $d_m^{(k)}$ is the m-step Arnoldi approximation for the error $\varepsilon_m^{(k)} = F(A)b - f_m^{(k)}$. Clearly, the preceding statement only makes sense if $\varepsilon_m^{(k)} = F^{(k+1)}(A)b^{(k+1)}$ is again the action of a matrix function $F^{(k+1)}(A)$ on a normalized vector $b^{(k+1)}$. Note that in general $F^{(k)} \neq F$, and, as it turns out, in all known approaches one finds $b^{(k)} = v_{m+1}^{(k-1)}$, the $(m+1)$st Arnoldi vector of the previous cycle. A generic algorithmic description is given in Algorithm 2.1.

Algorithm 2.1 Restarted Arnoldi method for $F(A)b$ from [25] as presented in [29].

1: Compute the Arnoldi decomposition $AV_m^{(1)} = V_m^{(1)}H_m^{(1)} + h_{m+1,m}^{(1)}v_{m+1}^{(1)}e_m^{\mathsf{T}}$
    with respect to A and b.
2: Set $f_m^{(1)} = \|b\|_2 V_m^{(1)} F(H_m^{(1)})e_1$.
3: **for** $k = 2, 3, \ldots$ until convergence **do**
4:     Determine the error function $F^{(k)}$ s.t. $\varepsilon^{(k-1)} = F^{(k)}(A)v_{m+1}^{(k-1)}$.
5:     Compute the Arnoldi decomposition $AV_m^{(k)} = V_m^{(k)}H_m^{(k)} + h_{m+1,m}^{(k)}v_{m+1}^{(k)}e_m^{\mathsf{T}}$
        with respect to A and $v_{m+1}^{(k-1)}$.
6:     Set $f_m^{(k)} = f_m^{(k-1)} + \|b\|_2 V_m^{(k)} F^{(k)}(H_m^{(k)})e_1$.
7: **end for**

In early work on restarting for matrix functions, $F^{(k)}, k > 1$ is characterized in

terms of divided differences of F. We have

$$F^{(2)}(s) = \|b\|_2 \prod_{i=1}^{m} h_{i+1,i}[D_{w_m}F](s),$$

where $[D_{w_m}F]$ denotes the m-th divided difference of F with respect to interpolation nodes which are the eigenvalues of $H_m^{(1)}$; see [25, 36, 48]. This expression can be iterated to obtain (iterated) divided difference representations for the error functions $F^{(k)}(t), k > 2$. The divided difference representations might be helpful to keep in mind when following the theory developed in this paper. They are, however, only marginally useful when it comes to computation, since they suffer from numerical instability.

This is why error function representations for rational functions in partial fraction form [2] and integral representations for the error based on Cauchy's integral formula for analytic functions [29] were developed as an alternative. These representations significantly improve numerical stability and robustness compared to earlier representations.

The following result from [29] is valid for *Stieltjes functions*, i.e., functions of the form

$$F(s) = \int_0^\infty \frac{1}{t+s} \, d\mu(t),$$

where μ is a nonnegative measure on the positive real axis such that $\int_0^\infty \frac{1}{t+1} \, d\mu(t) < \infty$. For many practically relevant Stieltjes functions, one has $d\mu(t) = \rho(t) \, dt$, i.e.,

$$(2.2) \qquad\qquad F(s) = \int_0^\infty \frac{\rho(t)}{t+s} \, dt,$$

where $\rho(t) \geq 0$ is a piecewise continuous function on $(0, \infty)$. Important examples of Stieltjes functions are $F(s) = s^{-\alpha}, \alpha \in (0, 1)$ and $F(s) = \log(1+s)/s$.

THEOREM 2.1 (adapted from Theorem 3.4 and Corollary 3.5 in [29]). *Let F be a function of the form* (2.2), *assume that* $\text{spec}(A) \cap (-\infty, 0] = \emptyset$, *denote by $f_m^{(k)}$ the restarted Arnoldi approximation for $F(A)b$ from the kth Arnoldi cycle and let $H_m^{(j)}, V_m^{(j)}, j = 1, \ldots, k$ be the Hessenberg matrix and orthonormal basis from the jth Arnoldi cycle. Let $\psi_m^{(j)}(t) = e_m^{\mathsf{T}}(H_m^{(j)} + tI)^{-1}e_1$. Then*[1]

$$F(A)b - f_m^{(k)} = (-1)^k \Big(\prod_{j=1}^{k} h_{m+1,m}^{(j)} \Big) \|b\|_2 \int_0^\infty \rho(t) \Big(\prod_{j=1}^{k} \psi_m^{(j)}(t) \Big)(A+tI)^{-1}v_{m+1}^{(k)} \, dt$$

$$=: F^{(k+1)}(A)v_{m+1}^{(k)}.$$

We will present an alternative proof to the one from [29] for Theorem 2.1 later in Corollary 3.5. Based on Theorem 2.1, a convergence analysis of the restarted Arnoldi process for Stieltjes functions was presented in [28]. From a computational perspective, the error representation given in Theorem 2.1 has the shortcoming that it cannot be evaluated by a closed formula. In [29], it is proposed to approximately evaluate $F^{(k)}$ by adaptive numerical quadrature. The crucial observation in this

---

[1]Note that the result as given in [29] has an erroneous factor of -1.

context is that one only needs to evaluate $F^{(k)}$ at the (small) Hessenberg matrix $H_m^{(k+1)}$, and not at A itself. Thus, when implemented carefully, the computational cost of this quadrature rule is negligible compared to matrix-vector products with A and orthogonalization costs. For an overview of several different, suitable quadrature rules tailored for specific practically relevant Stieltjes functions, we refer to [29, Section 4].

Remark 2.2. There is another, conceptually different way of performing restarts in the Arnoldi method, the so-called "residual-time restarting" developed in [13, 14], based on [12]. This approach is limited to functions like the exponential and φ-functions since it exploits the connection to an underlying ordinary differential equation. Instead of trying to approximate an *error function*, as outlined above, it is based on a *time-stepping approach*, where each restart cycle propagates the iterate from the previous cycle forward in time, until the desired point is reached.

2.2. The Laplace transform. The function given by the integral

$$(2.3) \qquad \mathcal{L}_t\{f(t)\}(s) = \int_0^\infty f(t)\exp(-st)\,\mathrm{d}t$$

is called the *Laplace transform of f*. Whenever the integration variable is clear, we will simply write $\mathcal{L}\{f\}(s)$. In this section we summarize the most important properties of Laplace transforms needed in this paper.

The set of all values $s \in \mathbb{C}$ for which $\mathcal{L}\{f\}(s)$ converges (absolutely) is called the *region of (absolute) convergence*. It generally has the shape of a half plane:

THEOREM 2.3 ([20, Theorem 3.1]). *If* (2.3) *converges absolutely at s_0, then it converges absolutely in the closed right half-plane* $\mathrm{Re}(s) \geq \mathrm{Re}(s_0)$.

The smallest value α such that $\mathcal{L}\{f\}(s)$ converges absolutely for $\mathrm{Re}(s) > \alpha$ is called the *abscissa of absolute convergence*. In the case of simple convergence, we obtain a (possibly open) right half-plane $\mathrm{Re}(s) > \alpha_0$ with $\alpha_0 \leq \alpha$ being called the abscissa of convergence. However, only considering absolute convergence is not a restriction in the sense of the following theorem:

THEOREM 2.4 ([20, Theorem 3.4]). *If* (2.3) *converges for s_0, then it converges in the open half-plane* $\mathrm{Re}(s) > \mathrm{Re}(s_0)$, *where it can be expressed by the absolutely converging Laplace transform*

$$\mathcal{L}\{f\}(s) = (s - s_0)\,\mathcal{L}\{\phi\}(s - s_0)$$

with

$$\phi(t) = \int_0^t \exp(-s_0\tau)f(\tau)\,\mathrm{d}t.$$

Within its region of convergence, a Laplace transform always represents an analytic function.

THEOREM 2.5 ([20, Theorem 6.1]). *Let $\mathcal{L}\{f\}(s)$ converge for $\mathrm{Re}(s) > \alpha_0$. Then all its derivatives exist for $\mathrm{Re}(s) > \alpha_0$, and they are Laplace transforms, too,*

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

Suppose we are given a function $F(s)$ and want to represent it as a Laplace transform. By Theorem 2.5 a necessary condition on F, such that $\mathcal{L}\{f\}(s) = F(s)$

exists, is that $F(s)$ itself is analytic in a region $\text{Re}(s) > \alpha_0$. Some publications have examined more precise conditions, see, e.g., [16, 20, 50, 51] and the Paley-Wiener theorem, e.g., in [41].

A sufficient condition for F to be a Laplace transform is that $F(s)$ is a Stieltjes function, see, e.g., [50, Theorem 4a]. This can easily be verified by exploiting that

$$\frac{1}{t+s} = \mathcal{L}_\tau\{\exp(-t\tau)\}(s), \quad \text{Re}(s) > -t,$$

so that we can rewrite (2.2) as

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

Since this condition is not necessary, see, e.g., [11], we know that the class of Stieltjes functions is a subclass of the class of all functions that allow a Laplace transform representation.

Krylov methods for $F(A)b$ where $F(z)$ is a Laplace transform (or, more generally, a *Laplace–Stieltjes function*) have been considered before. E.g., in [9], a "tensorized" Krylov method for efficiently approximating functions of certain Kronecker-structured matrices is proposed, while [38] discusses pole selection for the rational Krylov method for Laplace transforms. Remark 1 in [21] mentions that for Hermitian matrices the error ε_m of the Lanczos approximation decreases strictly monotonically if $F(z) = \mathcal{L}\{f\}(z)$ with real nonnegative f. This result was later generalized to extended Krylov subspace methods in [44]. However, no restart approach for general Laplace transforms has been developed so far.

3. Restarts for Laplace transforms: Theory. This section contains our main theoretical result, Theorem 3.2. It is built on a known representation of ε_m for $F(z) = \exp(-tz)$ that we restate in Lemma 3.1. We conclude the section by emphasizing the connection to the earlier work on Stieltjes functions in Corollary 3.5.

Suppose we are interested in the action of the exponential function,

$$y(t) = \exp(-tA)b.$$

The Arnoldi approximation in this case is given by

$$y_m(t) = \|b\|_2 V_m \exp(-tH_m)e_1.$$

We explicitly include the dependency on t, here, since we need it later. The following lemma gives an expression for the error $\varepsilon_m(t)$. This result has already been used, e.g., to obtain error bounds in [22, eq. (32)] for Hermitian A and in [49, Theorem 3.1] for non-Hermitian A. We repeat the proof here for convenience.

LEMMA 3.1. *The error* $\varepsilon_m(t) = y(t) - y_m(t)$ *can be written as*

$$\varepsilon_m(t) = -h_{m+1,m}\|b\|_2 \int_0^t \exp((\tau - t)A)v_{m+1}g(\tau)\,\mathrm{d}\tau$$

where

(3.1) $$g(\tau) = e_m^\mathsf{T} \exp(-\tau H_m)e_1$$

is the $(m, 1)$ *entry of* $\exp(-\tau H_m)$.

Proof. First note that

$$y'_m(t) = -\|b\|_2 V_m H_m \exp(-tH_m)e_1 = -Ay_m(t) + h_{m+1,m}\|b\|_2 e_m^\mathsf{T} \exp(-tH_m)e_1 v_{m+1},$$

where in the second equality we have used the Arnoldi relation (1.1). The derivative of the error thus satisfies

$$\varepsilon'_m(t) = y'(t) - y'_m(t) = -A\varepsilon_m(t) - h_{m+1,m}\|b\|_2 g(t)v_{m+1}.$$

With $\varepsilon_m(0) = 0$, this is an initial-value problem. The assertion then follows, e.g., by the variation of constants formula. □

Since the matrix Laplace transform can be expressed as an integral involving the matrix exponential, we can use Lemma 3.1 to develop a new error representation. The result is given in the following theorem in which we use the field of values $W(A)$ of a square matrix A defined as $W(A) = \{x^\mathsf{H} Ax : \|x\|_2 = 1\}$.

THEOREM 3.2. *Let $F(s) = \mathcal{L}\{f(t)\}(s)$ be the Laplace transform of $f(t)$. If*

$$\nu = \min_{\mu \in W(A)} \mathrm{Re}(\mu)$$

lies within the region of absolute convergence of $\mathcal{L}\{f(t)\}(s)$, then the error ε_m of the Arnoldi approximation can be represented via a matrix Laplace transform

$$\varepsilon_m = F(A)b - \|b\|_2 V_m F(H_m)e_1 = -h_{m+1,m}\|b\|_2 \mathcal{L}\{\tilde{f}\}(A)v_{m+1},$$

where

$$\tilde{f}(t) = \int_0^\infty f(t+\tau)g(\tau)\,\mathrm{d}\tau, \quad \text{with } g(\tau) = e_m^\mathsf{T} \exp(-\tau H_m)e_1 \text{ (see (3.1)).}$$

Moreover, ν lies in the region of absolute convergence of $\mathcal{L}\{\tilde{f}\}(s)$.

Proof. For simplicity and without loss of generality we assume $\|b\|_2 = 1$. Then

$$\varepsilon_m = \int_0^\infty f(t)(\exp(-tA)b - V_m \exp(-tH_m)e_1)\,\mathrm{d}t$$

and applying Lemma 3.1 gives

$$\varepsilon_m = -h_{m+1,m} \int_0^\infty f(t) \int_0^t \exp((s-t)A)v_{m+1}g(s)\,\mathrm{d}s\,\mathrm{d}t.$$

With the transformation $\tau = t - s$ in the inner integral we obtain

$$\varepsilon_m = -h_{m+1,m} \int_0^\infty \int_0^t f(t)\exp(-\tau A)v_{m+1}g(t-\tau)\,\mathrm{d}\tau\,\mathrm{d}t$$

$$(3.2) \qquad = -h_{m+1,m} \int_0^\infty \int_0^\infty f(t)\exp(-\tau A)v_{m+1}g(t-\tau)u(t-\tau)\,\mathrm{d}\tau\,\mathrm{d}t$$

with $u(x)$ being the Heaviside step function,

$$u(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases}.$$

Assume for now that we can interchange the order of integration. Then

$$\varepsilon_m = -h_{m+1,m} \int_0^\infty \exp(-\tau A) v_{m+1} \int_0^\infty f(t) g(t-\tau) u(t-\tau) \, dt \, d\tau$$

$$= -h_{m+1,m} \int_0^\infty \exp(-\tau A) v_{m+1} \int_\tau^\infty f(t) g(t-\tau) \, dt \, d\tau,$$

where

$$\int_\tau^\infty f(t) g(t-\tau) \, dt = \int_0^\infty f(t+\tau) g(t) \, dt = \tilde{f}(\tau),$$

which is the assertion of the theorem. That the order of integration in (3.2) can indeed be interchanged and that ν lies in the region of absolute convergence of $\mathcal{L}\{\tilde{f}\}$ is shown in Appendix A. □

Since the representation for ε_m uses again a matrix Laplace transform which converges absolutely for ν, we can apply Theorem 3.2 for later restarts, too. We summarize this as a corollary.

COROLLARY 3.3. *Let the assumptions of Theorem 3.2 hold. Define*

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

with $f^{(1)} = f$ and $g^{(j)}(\tau) = e_m^\mathsf{T} \exp(-\tau H_m^{(j)}) e_1$ for $j \geq 1$. Then the error $\varepsilon_m^{(k)}$ after $k \geq 1$ restart cycles satisfies

$$\varepsilon_m^{(k)} = (-1)^k \Big(\prod_{j=1}^k h_{m+1,m}^{(j)} \Big) \|b\|_2 \, \mathcal{L}\{f^{(k+1)}\}(A) v_{m+1}^{(k)}.$$

Remark 3.4. Let A be Hermitian and $F(z) = \mathcal{L}\{f\}(z)$ with real nonnegative $f = f^{(1)}$. Then the error $\varepsilon_m^{(1)}$ decreases strictly monotonically with growing m; see the discussion at the end of section 2. In this case we also have that $\tilde{f} = f^{(2)}$ has constant sign. Thus, by induction, the error also decreases monotonically within every following restart cycle.

Theorem 3.2 gives an error function representation for the restarted Arnoldi algorithm in terms of a Laplace transform. Before we move to possible extensions in section 4, we now show that the error representation for Stieltjes functions can be considered a special case of Corollary 3.3.

COROLLARY 3.5. *If $F(s)$ is a Stieltjes function, i.e.,*

$$F(s) = \mathcal{L}\{f\}(s) = \mathcal{L}\{\mathcal{L}\{\rho\}\}(s),$$

then the error function representation in Corollary 3.3 reduces to Theorem 2.1 in the sense that

$$\mathcal{L}\{f^{(k+1)}\}(A) v_{m+1}^{(k)} = \int_0^\infty \rho(t) \Big(\prod_{j=1}^k \psi_m^{(j)}(t) \Big) (A+tI)^{-1} v_{m+1}^{(k)} \, dt.$$

Proof. Due to the fact that every Stieltjes function is a double Laplace transform, we can rewrite the statement of the corollary as

$$\mathcal{L}\{f^{(k+1)}\}(A)v_{m+1}^{(k)} = \mathcal{L}^2\{\rho(t)\big(\prod_{j=1}^{k}\psi_m^{(j)}(t)\big)\}(A)v_{m+1}^{(k)},$$

and it is sufficient to show that

$$f^{(k+1)} = \mathcal{L}\{\rho(t)\big(\prod_{j=1}^{k}\psi_m^{(j)}(t)\big)\}.$$

We prove this by induction over $k \geq 0$: The case $k = 0$ is trivial, since by the corollary's hypothesis we have $f^{(1)} = \mathcal{L}\{\rho\}$. Now, assume that the above equation holds for $k - 1$, i.e., $f^{(k)} = \mathcal{L}\{\rho(t)\big(\prod_{j=1}^{k-1}\psi_m^{(j)}(t)\big)\}$. Then, by definition of $f^{(k+1)}$,

$$f^{(k+1)}(s) = \int_s^\infty f^{(k)}(t)g^{(k)}(t-s)\,\mathrm{d}t$$

$$= \mathcal{L}\{\mathcal{L}^{-1}\{f^{(k)}\}\,\mathcal{L}\{g^{(k)}\}\}(s) = \mathcal{L}\{\rho(t)\big(\prod_{j=1}^{k-1}\psi_m^{(j)}(t)\big)\,\mathcal{L}\{g^{(k)}\}\},$$

where for the second equality we used [4, Theorem 2.1]. What remains to show is that $\mathcal{L}\{g^{(k)}\} = \psi_m^{(k)}$ holds, which follows as

$$\mathcal{L}\{g^{(k)}\}(t) = e_m^\mathsf{T}\mathcal{L}_\tau\{\exp(-\tau H_m^{(k)})\}(t)e_1 = e_m^\mathsf{T}(tI + H_m^{(k)})^{-1}e_1 = \psi_m^{(k)}(t). \qquad \square$$

4. Extensions. The approach developed for Laplace transforms can easily be ported to two further classes of functions as we shortly discuss now.

4.1. Two-sided Laplace transforms. *Two-sided* Laplace transforms

$$\mathcal{B}\{f\}(s) = \int_{-\infty}^\infty f(t)\exp(-st)\,\mathrm{d}t = \mathcal{L}\{f(-t)\}(-s) + \mathcal{L}\{f(t)\}(s)$$

can be interpreted as the generalization of the Fourier transform

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

to complex arguments ω. The region of (absolute) convergence in the two-sided case is generally a vertical strip instead of a half-plane. Computationally, we can approximate $\mathcal{B}\{f\}(A)b$ using the Arnoldi approach for the two Laplace transforms $\mathcal{L}\{f(-t)\}(-A)$ and $\mathcal{L}\{f(t)\}(A)$. Note that the Krylov subspaces for both Laplace transforms are identical and need thus be built only once.

4.2. Complete Bernstein functions. A function $F : (0, \infty) \to [0, \infty)$ is called a *Bernstein function* if $F \in C^\infty$ and its derivative F' is completely monotone, i.e.,

$$\left(\frac{\mathrm{d}}{\mathrm{d}s}\right)^n (-1)^{n-1}F(s) \geq 0 \quad \text{for } s > 0 \,, \ n \in \mathbb{N}.$$

See [11, 43] for more information about Bernstein functions.

A Bernstein function $F(s)$ is called *complete* if it admits the representation

(4.1)
$$F(s) = c + as + \int_0^\infty (1 - \exp(-st))f(t)\,\mathrm{d}t$$

where $f(t)$ is completely monotone and $c, a \geq 0$.[2] If for a complete Bernstein function we compute $(cI + aA)b$ directly and use the Arnoldi approximation on $\mathcal{K}_m(A, b)$ for the integral in (4.1), then the error of the resulting approximation f_m is

$$F(A)b - f_m = \int_0^\infty (I - \exp(-tA))bf(t)\,\mathrm{d}t - \int_0^\infty V_m(I - \exp(-tH_m))e_1 f(t)\,\mathrm{d}t$$

(4.2)
$$= -\int_0^\infty (\exp(-tA)b - V_m \exp(-tH_m)e_1)f(t)\,\mathrm{d}t,$$

which is the error for the Laplace transform $-\mathcal{L}\{f\}$ provided it exists. Consequently, the error representation for Laplace transforms can also be used for complete Bernstein functions. Some care is required, though, regarding the regions of convergence, since at first sight it seems as if ν should lie in the region of absolute convergence of $\mathcal{L}\{f\}$ with f from (4.1), and this might be quite restrictive. However, an inspection of the proof of Theorem 3.2 given in Appendix A shows that it is sufficient to have that ν lies in the region of absolute convergence of $\mathcal{L}\{tf(t)\}$. Since from (4.1) we have

$$F'(s) = a + \int_0^\infty t \exp(-st)f(t)\,\mathrm{d}t = a + \mathcal{L}\{tf(t)\}(s),$$

this means that the error representation in Theorem 3.2 and Corollary 3.3 is also valid with converging integrals if F is a complete Bernstein function for which ν lies in the region of absolute convergence of F'.

5. Restarts for Laplace transforms: Implementation aspects. The representation of the error function in Theorem 3.2 enables us to design a new restarted Arnoldi algorithm for Laplace transforms: In Algorithm 2.1, we have to set

$$F^{(k)}(H_m^{(k)})e_1 = (-1)^{k-1}\Big(\prod_{j=1}^{k-1} h_{m+1,m}^{(j)}\Big)\|b\|_2\,\mathcal{L}\{f^{(k)}\}(H_m^{(k)})e_1,$$

see Corollary 3.3. The new algorithm is given in Algorithm 5.1, containing references to important aspects of implementation discussed in the remainder of this section.

5.1. Quadrature. To evaluate the error function, we apply a quadrature rule

(5.1)
$$\mathcal{L}\{f^{(k)}\}(H_m^{(k)})e_1 \approx \sum_{i=1}^\ell w_i f^{(k)}(t_i) \exp(-t_i H_m^{(k)})e_1.$$

Because of the term involving the exponential function, we expect that the main contributions to the integral come from the interval containing the smaller real parts

---

[2]In fact, every Bernstein function can be represented as $F(s) = c + as + \int_0^\infty (1 - \exp(-st))\,\mathrm{d}\mu(t)$ where μ is a positive measure satisfying $\int_0^\infty \frac{t}{t+1}\,\mathrm{d}\mu(t) < \infty$, see [43, Theorem 3.2]. Thus, the measure μ of a complete Bernstein function has a completely monotone density f with respect to the Lebesgue measure.

Algorithm 5.1 Restarted Arnoldi method for $F(A)b$ where F is a Laplace transform

---

1: Compute the Arnoldi decomposition $AV_m^{(1)} = V_m^{(1)} H_m^{(1)} + h_{m+1,m}^{(1)} v_{m+1}^{(1)} e_m^{\mathsf{T}}$
     with respect to A and b.
2: Set $f_m^{(1)} = \|b\|_2 V_m^{(1)} F(H_m^{(1)}) e_1$.
3: Choose a quadrature rule with nodes t_i and weights w_i, $i = 1, \ldots, \ell$, to evaluate
     the Laplace transform of f at $H_m^{(1)}$ (see subsection 5.1).
4: **for** $k = 2, 3, \ldots$ until convergence **do**
5:     Compute the Arnoldi decomposition $AV_m^{(k)} = V_m^{(k)} H_m^{(k)} + h_{m+1,m}^{(k)} v_{m+1}^{(k)} e_m^{\mathsf{T}}$
         with respect to A and $v_{m+1}^{(k-1)}$.
6:     Choose a quadrature rule with nodes t_i and weights w_i, $i = 1, \ldots, \ell$, to evaluate
         the Laplace transform of $f^{(k)}$ at $H_m^{(k)}$ (see subsection 5.1).
7:     **if** $k = 2$ **then**
8:         Define $s^{(k-1)} = f^{(k-1)}$.
9:     **else**
10:        Construct a spline $s^{(k-1)}$ that interpolates $f^{(k-1)}$.
11:    **end if**
12:    Approximate $(f^{(k)}(t_i))_{i=1,\ldots,\ell}$ as in (5.2) using $s^{(k-1)}$.
13:    Compute $d_m^{(k-1)} = (-1)^{k-1} \left(\prod_{j=1}^{k-1} h_{m+1,m}^{(j)} \right) \|b\|_2 V_m^{(k)} \mathcal{L}\{f^{(k)}\}(H_m^{(k)}) e_1$ by
         numerical quadrature (see (5.1)).
14:    Set $f_m^{(k)} = f_m^{(k-1)} + d_m^{(k-1)}$.
15: **end for**

---

of the eigenvalues of $H_m^{(k)}$. Consequently, we determine a quadrature rule based on the smallest real part of the spectrum of $H_m^{(1)}$: We set

$$\nu = \min_{x \in \mathrm{spec}(H_m^{(1)})} \mathrm{Re}(x)$$

and determine the nodes t_i and the weights w_i of the quadrature rule in (5.1) in the same way MATLAB's routine `integral` would evaluate the integral

$$\mathcal{L}\{f^{(1)}\}(\nu) = \int_0^\infty f^{(1)}(t) \exp(-t\nu) \, \mathrm{d}t$$

with relative and absolute target accuracy ε_{q}. This means that after the transformation $x = \sqrt{t}/(1+\sqrt{t})$, an adaptive Gauss-Kronrod scheme is applied to subintervals of the new integration interval $[0, 1)$; see [46] for more details. The nodes t_i and weights w_i obtained this way make up the quadrature rule (5.1) for the integral $\mathcal{L}\{f^{(k)}\}(H_m^{(k)})$; if necessary, after each restart, the rule can be computed anew and thus adapted to the new error representation. To simplify notation we do not explicitly denote this dependency of ℓ, t_i and w_i on k in Algorithm 5.1 and elsewhere. In principle, one might aim at developing and applying more advanced quadrature procedures to our specific situation, but the method just outlined was sufficient in all our experiments.

5.2. Spline interpolation. Since $f^{(k)}(t) = e_k^{\mathsf{T}} \mathcal{L}_\tau \{f^{(k-1)}(t+\tau)\}(H_m^{(k-1)}) e_1$ can be interpreted as another Laplace transform evaluated at the Hessenberg matrix $H_m^{(k)}$, we can use the quadrature rule from above to evaluate $f^{(k)}$, too. However, $f^{(k)}$ is a recursive integral. While it is possible to apply a series of quadrature rules to the

expanded expression

$$f^{(k)}(t) = \int_0^\infty \cdots \int_0^\infty f\left(t + \sum_{i=1}^{k-1} \tau_i\right) \prod_{i=1}^{k-1} g^{(i)}(\tau_i) \, d\tau_1 \ldots d\tau_{k-1},$$

this approach is prohibitively expensive except for very small k. Instead of the quadrature rule

$$f^{(k)}(t) \approx \sum_{i=1}^\ell w_i f^{(k-1)}(t + t_i) g^{(k-1)}(t_i),$$

we propose to use

(5.2) $$f^{(k)}(t) \approx \sum_{i=1}^\ell w_i s^{(k-1)}(t + t_i) g^{(k-1)}(t_i),$$

where $(t_i, w_i)_{i=1,\ldots,\ell}$ are the quadrature nodes and weights of subsection 5.1 and $s^{(k-1)}$ is a cubic spline interpolating $f^{(k-1)}$ at points $x_1 < x_2 < \cdots < x_q$ whose selection we will discuss below.

Recall that an interpolating cubic spline is a function of the form

$$s(x) = \begin{cases} p_1(x) & x \le x_2 \\ p_2(x) & x_2 \le x \le x_3 \\ \quad \vdots \\ p_{q-1}(x) & x_{q-1} \le x \end{cases},$$

where each p_i is a cubic polynomial. Note that we included extrapolation for $x < x_1$ and $x > x_q$ in the above formula. If $s(x)$ interpolates $f(x)$ at the q points x_i, $i = 1, \ldots, q$ then we have $s(x_i) = f(x_i)$ and s satisfies the smoothness conditions

$$p_{j-1}(x_j) = p_j(x_j),$$
$$p'_{j-1}(x_j) = p'_j(x_j),$$
$$p''_{j-1}(x_j) = p''_j(x_j),$$

with $j = 2, \ldots, q - 1$. Our implementation uses MATLAB's `spline` functionality which applies "not-a-knot" boundary conditions (meaning that $p_1 = p_2$ and $p_{q-2} = p_{q-1}$). For a detailed treatment of spline interpolation, see, e.g., [19].

We now discuss how to choose the interpolation nodes. In cycle $k - 1$, we have computed

$$\mathcal{L}\{f^{(k-1)}\}(H_m^{(k-1)})e_1 \approx \sum_{i=1}^\ell w_i f^{(k-1)}(t_i) \exp(-t_i H_m^{(k-1)})e_1,$$

see subsection 5.1. Consequently, the values $f^{(k-1)}(t_i)$ are already known in the next cycle k and can be used as interpolation points for $s^{(k-1)}$. In this case, the quadrature nodes and the interpolation nodes coincide: $q = \ell$ and $x_i = t_i$. However, we observed that sometimes additional interpolation points are needed to ensure overall convergence of $f_m^{(k)}$ to $F(A)b$. An adaptive way to obtain those is to add the points

$(x_i + x_{i+1})/2$ to the set of interpolation nodes[3] and repeat the process until a suitable stopping criterion is fulfilled. If d_r denotes the value of $d_m^{(k-1)}$ after refinement step r, a possible criterion is that

$$(5.3) \qquad \|d_r - d_{r-1}\|_2 \leq \varepsilon_s \|f_m^{(k-1)}\|_2$$

with ε_s sufficiently small. The number of necessary refinement steps can be predicted. For example, if $f^{(k-1)} \in C^4[x_1, x_n]$, then the error bound (see [7, eq. (1.8)])

$$|f^{(k-1)}(t) - s^{(k-1)}(t)| \leq c \, \Delta x_i{}^2 \max_j \Delta x_j{}^2 \|\frac{\mathrm{d}^4}{\mathrm{d}t^4} f^{(k-1)}\|_\infty \quad \text{for } x_i \leq t \leq x_{i+1}$$

is reduced by the factor 16 in each step. If the number of interpolation nodes needed is $\mathcal{O}(\ell^2)$, it is more advantageous to use $t_i + t_j$ with $i = 1, \ldots, \ell$, $j = i, \ldots, \ell$: In that case, we have

$$\sum_{i=1}^{\ell} w_i f^{(k-1)}(t+t_i) g^{(k-1)}(t_i) = \sum_{i=1}^{\ell} w_i s^{(k-1)}(t+t_i) g^{(k-1)}(t_i)$$

in (5.2) for each $t = t_i$ required by (5.1).

5.3. Matrix exponential function. In (5.1) and (5.2) we need to compute $\exp(-t_i H_m^{(k)}) e_1$ for several values of t_i. While MATLAB's `expm` is the state-of-the-art choice if the full matrix exponential is required, for our purposes we need an efficient approach to obtain the action of the matrix exponential on a vector, $\exp(-t_i H_m^{(k)}) e_1$, and this for several values t_i. If A is Hermitian, the Hessenberg matrices $H_m^{(k)}$ are Hermitian, too, and we use the eigendecomposition, i.e.,

$$\exp(-t_i H_m^{(k)}) e_1 = (X_m^{(k)})^{\mathsf{H}} \exp(-t_i D_m^{(k)})(X_m^{(k)} e_1)$$

where $X_m^{(k)}$ is the matrix of eigenvectors and $D_m^{(k)}$ is the diagonal matrix of eigenvalues of $H_m^{(k)}$. The eigendecomposition needs to be computed only once for all t_i. Since the eigendecomposition of a non-Hermitian matrix might involve non-trivial Jordan blocks or might be ill-conditioned, we take another approach for non-Hermitian A. We choose the MATLAB package `expmv`[4] which implements an algorithm described in [3]. It uses the scaling part of the scaling and squaring method and a truncated Taylor approximation together with some further preprocessing steps. The method avoids matrix-matrix products, so that generally it will compute $\exp(-t_i H_m^{(k)}) e_1$ faster than `expm`. For large t_i, however, the fact that the squaring part is missing in `expmv` makes `expm` faster. We modified `expmv` in our implementation to switch to `expm` in these cases.

6. Numerical experiments. We now present several numerical examples for the restarted Arnoldi method for matrix Laplace transforms.[5] All examples were calculated in MATLAB R2021a on a laptop with Intel® Core™ i7-8650U and 16 GB. We denote by `tol` the target relative error norm which was set to 10^{-7} in all our examples. To be on the safe side, the target accuracy for the quadrature rule ε_q was

---

[3]We assume that $f^{(k-1)}$ is approximated again as in (5.2) for these additional evaluations.
[4]available at https://github.com/higham/expmv
[5]source code available at https://github.com/MaTso7/laplace_restarting

chosen as $\varepsilon_{\mathrm{q}} = 10^{-3}\texttt{tol}$, although we observed that larger values up to $\texttt{tol}$ usually worked as well. The splines $s^{(k)}$ were refined as described in subsection 5.2 with $\varepsilon_{\mathrm{s}} = \varepsilon_{\mathrm{q}}$ in (5.3). For comparisons, we also used the MATLAB package $\texttt{funm\_quad}^6$ [30] that implements the algorithm of [29], where we chose the target accuracy for numerical integration to be $\texttt{tol}$ and, when applicable, the *two-pass Lanczos method* which can be used as an alternative to restarting in the Hermitian case. This method runs the Lanczos process twice (once for assembling the tridiagonal matrix H_m, once for forming the approximation f_m as linear combination of the basis vectors). This way, it is not necessary to store the full Krylov basis, at the price of roughly doubling the computational cost; see, e.g., [31] for details.

Remark 6.1. When comparing the performance of restarted methods and of two-pass Lanczos, one has to keep in mind that just counting matrix-vector products does not give the full picture of the computational cost of two-pass Lanczos: In order to monitor convergence, one needs to evaluate some stopping criterion (e.g., the norm of the difference between iterates), which at iteration j typically involves (at least) forming $F(H_j)$. If a large number of iterations is necessary, this induces a non-negligible additional cost, even if the stopping criterion is not checked after each iteration. In contrast, in a restarted method, one can compare the norm of iterates from subsequent cycles by only evaluating $F^{(k)}$ on matrices of the fixed size m. To make the methods comparable at least to some extent, we check for convergence every m iterations, where m is the restart length.

6.1. Fractional negative power less than -1: $F(s) = s^{-3/2}$. As a first example, consider

$$F(s) = s^{-3/2} = \frac{2}{\pi}\,\mathcal{L}\{\sqrt{t}\}(s).$$

This Laplace transform converges absolutely for $\mathrm{Re}(s) > 0$. Since $G(s) = s^{-1/2}$ is a Stieltjes function and $F(s) = G(s)s^{-1}$, we can compute $F(A)b$ by first solving the linear system $c = A^{-1}b$ and then treating $G(A)(A^{-1}b) = G(A)c$ as a Stieltjes function. This gives us an established method to compare Algorithm 5.1 against, even though the matrix function $F(A) = A^{-3/2}$ does not directly fit into any of the restart frameworks considered in the literature so far. For the action of the inverse, $A^{-1}b$, we use the MATLAB built-in implementations of CG when A is symmetric positive definite and restarted GMRES otherwise.

As matrix A, we first consider the discretized 3D Laplacian on a cubic grid with constant step size and Dirichlet boundary conditions, i.e.,

$$A_{\mathrm{L}} = A_1 \oplus A_1 \oplus A_1 \in \mathbb{R}^{N^3 \times N^3}.$$

Here $A_1 = \mathrm{tridiag}(-1, 2, -1) \in \mathbb{R}^{N \times N}$ denotes the 1D Laplace operator and $\oplus$ the Kronecker sum defined via the Kronecker product $\otimes$ as

$$M_1 \oplus M_2 = M_1 \otimes I_{m_2} + I_{m_1} \otimes M_2, \quad M_1 \in \mathbb{C}^{m_1 \times m_1}, M_2 \in \mathbb{C}^{m_2 \times m_2}.$$

As a non-Hermitian example, we take the convection-diffusion operator on the unit cube with the direction vector $[1, -1, 1]$ and a diffusion coefficient of $\epsilon = 10^{-3}$. Using
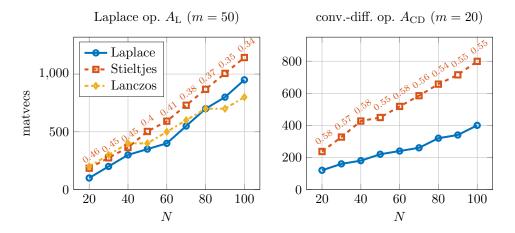

FIGURE 6.1. *Number of matrix-vector products for approximating* $A^{-3/2}b$. *"Laplace" denotes Algorithm 5.1, "Stieltjes" is the combination of* `funm_quad` *with CG (left) or GMRES (right). The small numbers above the dashed line indicate which fraction of the overall number of matrix-vector products was computed in the first phase, i.e., in the CG or GMRES method. "Lanczos" denotes the two-pass Lanczos method. The restart length is m.*

first-order upwind discretization for the convection term then results in the matrix

$$A_{\mathrm{CD}} = h^{-2}\epsilon A_{\mathrm{L}} + h^{-1}(A_2 \oplus A_2^\mathsf{T} \oplus A_2) \in \mathbb{R}^{N^3 \times N^3}$$

with $h = (N+1)^{-1}$ and $A_2 = \mathrm{tridiag}(-1,1,0) \in \mathbb{R}^{N \times N}$. Note that both A_{L} and A_{CD} are positive definite.

In our experiment, we vary N from 20 to 100, which gives matrix sizes between 8,000 and 1,000,000. We use a restart length of $m = 50$ for A_{L} and $m = 20$ for A_{CD}. We stop the iteration at the end of the first restart cycle for which the relative error norm lies below the desired accuracy $\mathtt{tol} = 10^{-7}$. To make sure that the solution of $A^{-1}b$ is sufficiently precise for this, we chose a target accuracy of 10^{-9} for the residual norm of CG and GMRES. For both methods, Figure 6.1 reports the required number of matrix-vector products for the different matrix sizes. We observe that the new restart approach requires significantly fewer matrix-vector products than the Stieltjes function approach. This is due to the fact that the "first phase", in which the CG or GMRES method is executed, is not necessary. To stress this effect further, we display the percentage of matrix-vector products that the first phase requires above each data point. We deduce that the restarted matrix function algorithms perform quite similarly and in fact, the Stieltjes function restarting method from [29] even requires a slightly smaller number of matrix-vector products. The advantage of the new method can thus almost completely be attributed to the fact that it directly approximates the desired quantity without needing to additionally solve a linear system first. For the symmetric matrix A_L, when comparing with the two-pass Lanczos approach, we see that the new approach requires fewer matrix-vector multiplications except for the two largest values of N.

To illustrate that the reduction in matrix-vector products does not come with a decrease in accuracy, we also depict the final relative accuracies in Figure 6.1 in Figure 6.2 as well as an exemplary convergence curve (for the largest problem instance $N = 100$) in Figure 6.3. The convergence curve only covers the second phase of
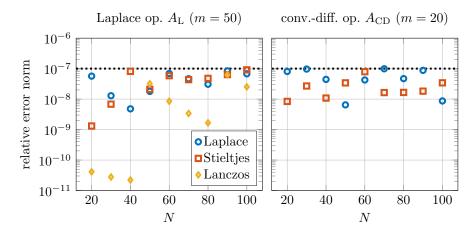
FIGURE 6.2. *Accuracy at termination when approximating $A^{-3/2}b$. "Laplace" denotes Algorithm 5.1, "Stieltjes" is the combination of `funm_quad` with CG (left) or GMRES (right). "Lanczos" denotes the two-pass Lanczos method. The restart length is m.*
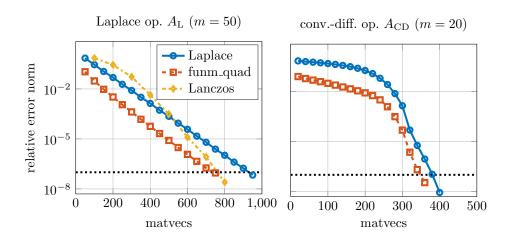

FIGURE 6.3. *Convergence curves for approximating $A^{-3/2}b$ with Algorithm 5.1 ("Laplace") and for approximating $A^{-1/2}c$ where $c = A^{-1}b$, with `funm_quad`. "Lanczos" denotes the two-pass Lanczos method. $N = 100$. The restart length is m.*

$A^{-1/2}(A^{-1}b)$, i.e., the approximation of the matrix function $A^{-1/2}$ applied to $c = A^{-1}b$. We observe that the error reduction rate of both restarted methods is very similar, but that `funm_quad` starts at a lower initial error norm so that it requires fewer restart cycles to reach the target accuracy. This is probably related to the different starting vectors b and $A^{-1}b$ in both methods. For A_L, non-restarted two-pass Lanczos exhibits superlinear convergence which would make this method require the least number of matrix-vector multiplications for accuracies beyond 10^{-7}.

Figure 6.4 illustrates that the lower number of matrix-vector products also translates into an advantage in run time, in particular for the larger problem instances. The overhead due to the spline construction and evaluation in Algorithm 5.1 (and due to the matrix exponential for $A = A_{CD}$) is relatively large for the smaller problem sizes,
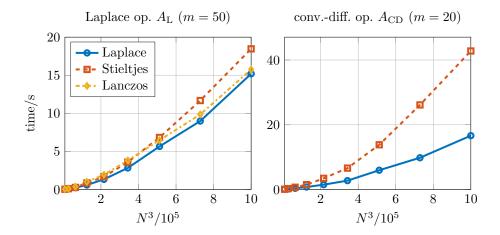
FIGURE 6.4.   *Execution times when approximating $A^{-3/2}b$ for varying matrix sizes N^3. "Laplace" denotes Algorithm 5.1, "Stieltjes" is the combination of* `funm_quad` *with CG (left) or GMRES (right). "Lanczos" denotes the two-pass Lanczos method. The restart length is m.*

so we do not observe a significant speed-up. For larger N, however, the speed-up is similar to what we would expect from Figure 6.1. Moreover, we see linear dependence on the matrix size N^3, i.e., the overhead is negligible for larger matrices, the cost of the methods being completely dominated by matrix-vector products. Two-pass Lanczos does not yet exhibit the overhead as discussed in Remark 6.1, probably because the solution is found at a rather moderate Krylov subspace size of about 400.

The results hold for other restart lengths m, too. As a small demonstration, we report the run times for varying restart lengths for $N = 100$ in Figure 6.5. For A_L, we had to include an additional factor of 10^{-2} in the target accuracy of the numerical integration in both Algorithm 5.1 and `funm_quad` for $m = 10$ and a factor of 10^{-1} in `funm_quad` for $m = 20, 30, 40$. While a larger m can possibly improve convergence and thus decrease the run time, this was only observed for the Hermitian case A_L. The increasing cost of orthogonalization for the next Arnoldi vector in the non-Hermitian case A_{CD} leads to an overall increasing execution time. In any case, Figure 6.5 shows that Algorithm 5.1 behaves similarly to other restart algorithms when changing m.

6.2. Fractional diffusion processes on graphs: $F(s) = \exp(-\tau\sqrt{s})$**.** Given an (undirected) graph G, its symmetric positive semidefinite *graph Laplacian* $L_G = D_G - A_G$, where A_G is the adjacency matrix and D_G the diagonal matrix containing the degrees of the nodes of G, the ODE system

$$(6.1) \qquad \begin{cases} \frac{d}{d\tau}u(\tau) &= -L_G^\alpha u(\tau), \quad \tau \in (0, T], \\ u(0) &= u_0 \end{cases}$$

(where $\alpha \in (0, 1)$) models a fractional diffusion process on G; see, e.g., [10]. Clearly, the solution of (6.1) is given by

$$(6.2) \qquad\qquad u(t) = \exp(-\tau L_G^\alpha)u_0.$$

It is well-known that the function $\exp(-\tau s^\alpha)$ appearing in (6.2) is a Laplace–Stieltjes function, as it is the composition of a Bernstein function and a completely monotonic
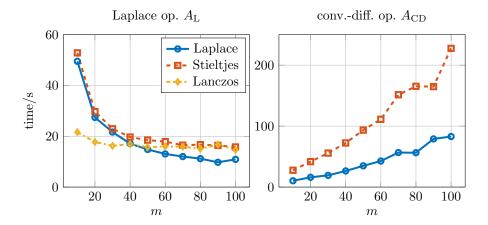
FIGURE 6.5. *Execution times when approximating $A^{-3/2}b$ for varying restart lengths m and $N = 100$. "Laplace" denotes* Algorithm *5.1, "Stieltjes" is the combination of* funm_quad *with CG (left) or GMRES (right). "Lanczos" denotes the two-pass Lanczos method. Note that for two-pass Lanczos, m specifies once in how many iterations the stopping criterion is checked.*

function. For $\alpha = 1/2$, we can give a closed form expression as a Laplace transform,

$$\exp(-\tau\sqrt{s}) = \frac{\tau}{2\sqrt{\pi}} \mathcal{L}\{\frac{\exp(-\tau^2/(4t))}{t^{3/2}}\}(s),$$

which converges absolutely for $\mathrm{Re}(s) \geq 0$, so that Algorithm 5.1 can be applied. For invertible A, it is sometimes proposed to write

(6.3) $$\exp(-\tau\sqrt{A})u_0 = (h(A)A + I)u_0$$

with the function $h(s) = (\exp(-\tau\sqrt{s}) - 1)/s$, which has the integral representation

(6.4) $$h(s) = -\int_0^\infty \frac{1}{s+t} \frac{\sin(\tau\sqrt{t})}{\pi t} \, dt;$$

see, e.g., [24]. While h is not a Stieltjes function in the strict sense due to the oscillating generating function in (6.4), the quadrature-based restarting technique from [29] is still applicable to it[7], thus giving an alternative way to approximate $\exp(-\tau\sqrt{A})$. However, in the context of fractional diffusion on graphs, it is worth noting that L_G is always a singular matrix so that $h(L_G)$ is not defined. As $L_G u_0$ does not contain any contribution from the nullspace of L_G, it is possible to define the action $h(L_G)L_G u_0$ by restriction to the orthogonal complement of the nullspace, but this makes some theoretical considerations more complicated. This holds in particular in the presence of round-off errors which may introduce (small) nullspace components during the Krylov iteration. Therefore, working directly with $\exp(-\tau\sqrt{s})$ is an attractive approach in this setting.

We used four real-world graphs for our experiments. The adjacency matrices were obtained from the SuiteSparse Matrix Collection [18]. For each graph, we extracted

---

[7]By using the Abel–Dirichlet test [47, Theorem 11.23a], one can easily check that also in this situation the integral representations of all occurring error functions are guaranteed to exist and have a finite value.

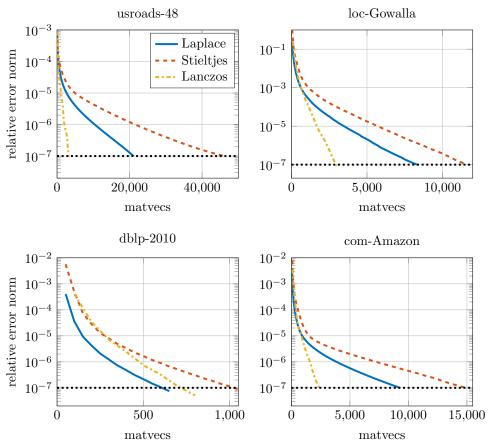| Name	nodes	edges
usroads-48	126 146	323 900
loc-Gowalla	196 591	1 900 654
dblp-2010	226 413	1 432 920
com-Amazon	334 863	1 851 744

FIGURE 6.6. *Convergence curves for approximating* $\exp(-\sqrt{A})b$. *"Laplace" denotes Algorithm* 5.1, *"Stieltjes" refers to the final error obtained by using* `funm_quad` *for* $h(A)(Ab)$ *in* $\exp(-\sqrt{A})b = (h(A)A + I)b$. *"Lanczos" denotes the two-pass Lanczos method. The restart length is* $m = 50$.

the largest connected component and only considered the corresponding Laplacian. Table 6.1 contains the number of nodes and edges of the resulting graphs. For sake of simplicity, we chose $\tau = 1$. The restart length is $m = 50$. We present the convergence curves in Figure 6.6. They show that in addition to the above argument, Algorithm 5.1 yields higher accuracy than the Stieltjes-type approach for the same number of matrix-vector products for each considered graph. Two-pass Lanczos requires the least number of matrix-vector multiplications except for the dblp-2010

graph.

Employing the "corrected" Arnoldi approximation introduced in [42], we can modify the restarting method for Stieltjes functions h to directly approximate $Ah(A)b$ without needing an explicit pre- or postmultiplication with A; see Corollary 3.6 and the discussion following it in [29]. We could apply this here, based on (6.3). For all graphs considered, however, this approach shows actually worse performance than the Stieltjes restart method applied to the starting vector $L_G b$. It sometimes even stagnates or shows signs of instability, which we attribute to the fact that L_G is singular. Thus premultiplying with L_G is actually beneficial, removing unwanted nullspace contributions from b.

6.3. Gamma function: $F(s) = \Gamma(s)$**.** Recently, techniques to compute the matrix gamma function $\Gamma(A)$ were discussed in [15, 39]. The gamma function can be represented as a two-sided Laplace transform,

$$\Gamma(s) = \int_0^\infty x^{s-1} \exp(-x)\, \mathrm{d}x = \mathcal{B}\{\exp(-\exp(-t))\}(s),$$

which converges absolutely for $\mathrm{Re}(s) > 0$. To see the last equality, use $x^{s-1} = \exp((s-1)\log(x))$ and apply the transformation $t = -\log(x)$. We include this example to show that the approach presented in this paper also works for the two-sided case. We compute $F(A)b$ by applying the method to the two Laplace transforms

$$F(A)b = \mathcal{L}\{\exp(-\exp(-t))\}(A)b + \mathcal{L}\{\exp(-\exp(t))\}(-A)b.$$

To determine the error, we implemented Algorithm 4.5 combined with 4.1 of [15] to compute $\Gamma(A)$ directly. As this becomes prohibitively expensive for larger matrices, we use the 2D versions of the matrices in subsection 6.1, i.e.,

$$A_\mathrm{L} = A_1 \oplus A_1 \in \mathbb{R}^{N^2 \times N^2},$$
$$A_\mathrm{CD} = h^{-2}\epsilon A_\mathrm{L} + h^{-1}(A_2 \oplus A_2^\mathsf{T}) \in \mathbb{R}^{N^2 \times N^2},$$

which are again positive definite. We let N vary from 20 to 120.

The number of matrix-vector products required is plotted in Figure 6.7 and we present the achieved accuracy in Figure 6.8. For the symmetric matrix A_L, two-pass Lanczos requires fewer matrix-vector multiplications for $N \geq 80$.

6.4. Square root: $F(s) = \sqrt{s}$**.** The square root is a complete Bernstein function since it can be represented as

$$F(s) = \sqrt{s} = \frac{1}{2\sqrt{\pi}} \int_0^\infty (1 - \exp(-st))t^{-3/2}\, \mathrm{d}t.$$

The corresponding action $F(A)b = A^{1/2}b$ arises in several applications, e.g., machine learning [40], sampling from Gaussian Markov random fields [36] and preconditioning [5]. Since

$$F'(s) = \frac{1}{2\sqrt{\pi}}\, \mathcal{L}\{t^{-1/2}\}(s)$$

converges absolutely for $\mathrm{Re}(s) > 0$, we can use the error representation of Theorem 3.2 for positive definite matrices, see subsection 4.2. For comparison, we use the package
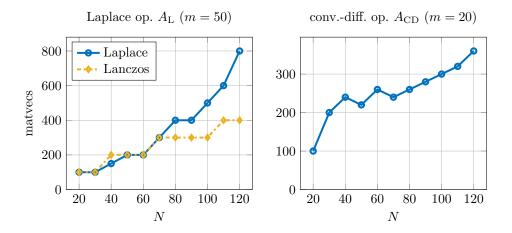
FIGURE 6.7. *Number of matrix-vector products for approximating* $\Gamma(A)b$. *The restart length is* m. *"Laplace" denotes Algorithm* 5.1. *"Lanczos" denotes the two-pass Lanczos method.*
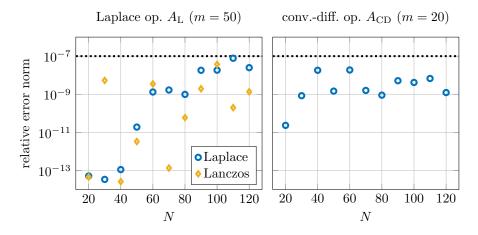

FIGURE 6.8. *Accuracy at termination when approximating* $\Gamma(A)b$. *"Laplace" denotes Algorithm* 5.1. *"Lanczos" denotes the two-pass Lanczos method. The restart length is* m.

`funm_quad` once again: Since $G(s) = s^{-1/2}$ is a Stieltjes function and $F(s) = G(s)s$, we can evaluate $F(A)b$ as $G(A)c$ with $c = Ab$. We choose again the Laplace operator A_L and the convection-diffusion operator A_{CD} from subsection 6.1. The number of matrix-vector products is plotted in Figure 6.9 and the resulting error in Figure 6.10. Both plots show once more that Algorithm 5.1 needs fewer matrix-vector products, with this time a more pronounced effect in the Hermitian case, where also two-pass Lanczos requires more matrix-vector multiplications (except for $N = 90$).

7. Conclusions. We developed a new representation of the error function in the restarted Arnoldi method for Laplace transforms. This representation allowed us to develop a new restart algorithm based on numerical integration and spline interpolation which significantly extends the class of functions for which restarting is possible in a black-box fashion without having to choose hand-tailored contours and
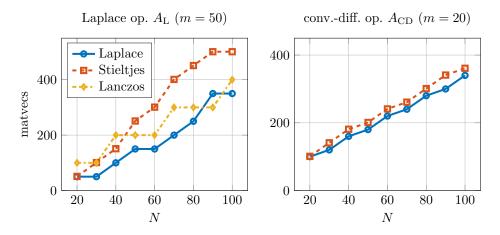
FIGURE 6.9. *Number of matrix-vector products for approximating $A^{1/2}b$. "Laplace" denotes Algorithm 5.1, "Stieltjes" is `funm_quad` for $A^{-1/2}c$ with $c = Ab$. "Lanczos" denotes the two-pass Lanczos method. The restart length is m.*
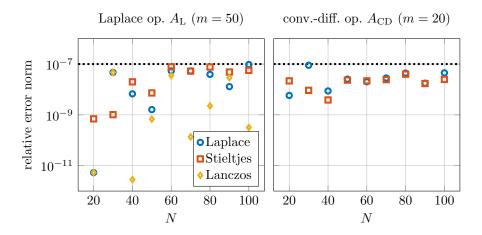

FIGURE 6.10. *Accuracy at termination when approximating $A^{1/2}b$. "Laplace" denotes Algorithm 5.1, "Stieltjes" is `funm_quad` for $A^{-1/2}c$ with $c = Ab$. "Lanczos" denotes the two-pass Lanczos method. The restart length is m.*

quadrature rules depending both on A and F.

Due to the results in [21], we know that the error monotonically decreases with our algorithm if A is Hermitian and f is nonnegative. We proposed an implementation which ensures that the run-time is dominated by matrix-vector products for larger matrices, and this has been verified in numerical experiments. Some of the newly available functions are not Stieltjes functions but can still be treated with the algorithm of [29] by multiplying with A or A^{-1}. However, our experiments also illustrated that the algorithm needs fewer matrix-vector products for similar accuracy in these cases. For the Hermitian examples, the two-pass Lanczos approach usually—but not always—required fewer matrix-vector multiplications than the new, restarted method. Lastly, we demonstrated that our method can be applied to two-sided Laplace trans-

forms, like the Gamma function, and complete Bernstein functions, like the square root, too.

Acknowledgments. We would like to thank Leonid Knizhnerman and an anonymous referee for their constructive comments which helped improve an earlier version of the manuscript.

Appendix A. This appendix shows that we can interchange the order of integration in (3.2), i.e., in the integral

$$\int_0^\infty \int_0^\infty f(t) \exp(-\tau A) v_{m+1} g(t-\tau) u(t-\tau) \, \mathrm{d}\tau \, \mathrm{d}t,$$

and that if ν lies within the region of absolute convergence of $\mathcal{L}\{f\}$, it also lies in the region of absolute convergence of $\mathcal{L}\{\tilde{f}\}$, which is what we left out in the proof of Theorem 3.2. Fubini's theorem (see, e.g., [41, Theorem 8.8]) states that we can interchange the order if

$$\mathcal{I} = \int_0^\infty \int_0^\infty |f(t) \exp(-\tau A) v_{m+1} g(t-\tau) u(t-\tau)| \, \mathrm{d}\tau \, \mathrm{d}t < \infty.$$

Note that the integral $\mathcal{I}$ and the integrand are vectors, so we have to show $\mathcal{I}_i < \infty$ for each entry $\mathcal{I}_i$ of $\mathcal{I}$. However, we can treat each entry in the same manner: For every vector $x \in \mathbb{C}^n$, it holds true that $|x_i| \leq \|x\|_2$. We apply this to the integrand in $\mathcal{I}$ so that we have

$$\mathcal{I}_i \leq \int_0^\infty \int_0^\infty \|\exp(-\tau A) v_{m+1}\|_2 \, |f(t)| \, |g(t-\tau)| \, u(t-\tau) \, \mathrm{d}\tau \, \mathrm{d}t$$

$$\leq \int_0^\infty \int_0^t \|\exp(-\tau A)\|_2 \, |f(t)| \, |g(t-\tau)| \, \mathrm{d}\tau \, \mathrm{d}t$$

for every $i = 1, \ldots, n$. Similarly, we use

$$|g(t)| = |e_m^\mathsf{T} \exp(-t H_m) e_1| \leq \|\exp(-t H_m) e_1\|_2 \leq \|\exp(-t H_m)\|_2$$

which results in

$$(\mathrm{A.1}) \qquad \mathcal{I}_i \leq \int_0^\infty \int_0^t \|\exp(-\tau A)\|_2 \, |f(t)| \, \|\exp(-(t-\tau) H_m)\|_2 \, \mathrm{d}\tau \, \mathrm{d}t.$$

Both matrix norms $\|\exp(-\tau A)\|_2$ and $\|\exp(-(t-\tau) H_m)\|_2$ can now be bounded by the result of Crouzeix and Palencia [17, Theorem 3.1]. We start with

$$\|\exp(-\tau A)\|_2 \leq c \max_{z \in W(A)} |\exp(-\tau z)| = c \max_{z \in W(A)} \exp(-\tau \operatorname{Re}(z)) = c \exp(-\tau \nu)$$

with the constant $c = 1 + \sqrt{2}$ and $\nu = \min_{z \in W(A)} \operatorname{Re}(z)$, where $W(A)$ is the field of values of A. Since $W(H_m) \subseteq W(A)$ and $t - \tau \geq 0$, one can proceed similarly for the second matrix norm, which yields

$$\|\exp(-(t-\tau) H_m)\|_2 \leq c \exp(-(t-\tau)\nu).$$

Using both bounds in (A.1), we obtain

$$
\begin{aligned}
\mathcal{I}_i &\leq \int_0^\infty \int_0^t c^2 \exp(-\tau\nu)|f(t)| \exp(-(t-\tau)\nu)\,\mathrm{d}\tau\,\mathrm{d}t \\
&= c^2 \int_0^\infty \exp(-t\nu)|f(t)| \int_0^t \mathrm{d}\tau\,\mathrm{d}t \\
&= c^2\,\mathcal{L}\{t|f(t)|\}(\nu),
\end{aligned}
$$

where the Laplace transform $-\mathcal{L}\{t|f(t)|\}(\nu)$ is the derivative of $\mathcal{L}\{|f|\}(\nu)$ according to Theorem 2.5. By the hypothesis of Theorem 3.2, the value ν lies within the region of absolute convergence of $\mathcal{L}\{f(t)\}(z)$. Thus, $-\mathcal{L}\{t|f(t)|\}(\nu)$ is finite and, consequently,

$$
\mathcal{I}_i \leq c^2\,\mathcal{L}\{t|f(t)|\}(\nu) < \infty.
$$

In an analogous manner, one shows that

$$
\mathcal{L}\{|\tilde{f}|\}(\nu) \leq c\,\mathcal{L}\{t|f(t)|\}(\nu) < \infty,
$$

which means that ν lies in the region of absolute convergence of $\mathcal{L}\{\tilde{f}\}$.

REFERENCES

[1] M. Afanasjew, M. Eiermann, O. G. Ernst, and S. Güttel, *A generalization of the steepest descent method for matrix functions*, Electron. Trans. Numer. Anal., 28 (2008), pp. 206–222.

[2] 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, https://doi.org/10.1016/j.laa.2008.06.029.

[3] A. H. Al-Mohy and N. J. Higham, *Computing the action of the matrix exponential, with an application to exponential integrators*, SIAM J. Sci. Comput., 33 (2011), pp. 488–511, https://doi.org/10.1137/100788860.

[4] R. AlAhmad, *Laplace transform of the product of two functions*, Italian J. Pure Appl. Math, 44 (2020), pp. 800–804, https://ijpam.uniud.it/online_issue/202044/71%20RamiAlahmad.pdf.

[5] M. Arioli and D. Loghin, *Discrete interpolation norms with applications*, SIAM J. Numer. Anal., 47 (2009), pp. 2924–2951.

[6] W. E. Arnoldi, *The principle of minimized iteration in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29, https://www.jstor.org/stable/43633863.

[7] R. K. Beatson, *On the convergence of some cubic spline interpolation schemes*, SIAM J. Numer. Anal., 23 (1986), pp. 903–912, https://doi.org/10.1137/0723058.

[8] M. Benzi and P. Boito, *Matrix functions in network analysis*, GAMM-Mitt., 43 (2020), p. e202000012, https://doi.org/10.1002/gamm.202000012.

[9] M. Benzi and V. Simoncini, *Approximation of functions of large matrices with Kronecker structure*, Numer. Math., 135 (2017), pp. 1–26, https://doi.org/10.1007/s00211-016-0799-9.

[10] M. Benzi and I. Simunec, *Rational Krylov methods for fractional diffusion problems on graphs*, BIT, 62 (2022), pp. 357–285, https://doi.org/10.1007/s10543-021-00881-0.

[11] 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.

[12] 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, https://doi.org/10.1137/110820191.

[13] M. A. Botchev, L. 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, https://doi.org/10.1137/20M1375383.

[14] 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, https://doi.org/10.1016/j.cam.2019.06.027.

[15] J. R. CARDOSO AND A. SADEGHI, *Computation of matrix gamma function*, BIT, 59 (2019), pp. 343–370, https://doi.org/10.1007/s10543-018-00744-1.

[16] J. L. B. COOPER, *The representation of functions as Laplace transforms*, Math. Ann., 159 (1965), pp. 223–233, https://doi.org/10.1007/BF01362440.

[17] 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, https://doi.org/10.1137/17M1116672.

[18] T. A. DAVIS AND Y. HU, *The University of Florida Sparse Matrix Collection*, ACM Trans. Math. Software, 38 (2011), https://doi.org/10.1145/2049662.2049663.

[19] C. DE BOOR, *A practical guide to splines*, Springer, New York, NY, 2001.

[20] G. DOETSCH, *Introduction to the Theory and Application of the Laplace Transformation*, Springer, Berlin, Germany, 1974, https://doi.org/10.1007/978-3-642-65690-3.

[21] V. DRUSKIN, *On monotonicity of the Lanczos approximation to the matrix exponential*, Linear Algebra Appl., 429 (2008), pp. 1679–1683, https://doi.org/10.1016/j.laa.2008.04.046.

[22] 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, https://doi.org/10.1137/S1064827596303661.

[23] V. DRUSKIN AND L. KNIZHNERMAN, *Two polynomial methods of calculating functions of symmetric matrices*, USSR Comput. Math. Math. Phys., 29 (1989), pp. 112–121, https://doi.org/10.1016/S0041-5553(89)80020-5.

[24] 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, https://doi.org/10.1137/S0895479895292400.

[25] 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, https://doi.org/10.1137/050633846.

[26] 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, https://doi.org/10.1016/S0010-4655(02)00455-1.

[27] J. VAN DEN ESHOF AND M. HOCHBRUCK, *Preconditioning Lanczos approximations to the matrix exponential*, SIAM J. Sci. Comput., 27 (2006), pp. 1438–1457, https://doi.org/10.1137/040605461.

[28] 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, https://doi.org/10.1137/140973463.

[29] 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, https://doi.org/10.1137/13093491X.

[30] A. FROMMER, S. GÜTTEL, AND M. SCHWEITZER, *FUNM_QUAD: An implementation of a stable, quadrature-based restarted Arnoldi method for matrix functions*, tech. report, Bergische Universität Wuppertal, 2014.

[31] 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.

[32] S. GÜTTEL, *Rational Krylov Methods for Operator Functions*, PhD thesis, Fakultät für Mathematik und Informatik der Technischen Universität Bergakademie Freiberg, 2010.

[33] 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, https://doi.org/10.1137/20M1351072.

[34] M. R. HESTENES AND E. STIEFEL, *Methods of conjugate gradients for solving linear systems*, J. Res. Natl. Bur. Stand., 49 (1952), pp. 409–436, https://doi.org/10.6028/jres.049.044.

[35] M. HOCHBRUCK AND A. OSTERMANN, *Exponential integrators*, Acta Numer., 19 (2010), pp. 209–286, https://doi.org/10.1017/S0962492910000048.

[36] 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, https://doi.org/10.1093/imanum/drp003.

[37] C. LANCZOS, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Natl. Bur. Stand., 45 (1950), pp. 255–282, https://doi.org/10.6028/jres.045.026.

[38] S. MASSEI AND L. ROBOL, *Rational Krylov for Stieltjes matrix functions: convergence and pole*

          selection, BIT, 61 (2021), pp. 237–273, https://doi.org/10.1007/s10543-020-00826-z.

[39] S. Miyajima, *Verified computation of matrix gamma function*, Linear Multilinear Algebra,
          (2020), https://doi.org/10.1080/03081087.2020.1757602.

[40] G. Pleiss, M. Jankowiak, D. Eriksson, A. Damle, and J. Gardner, *Fast matrix square
          roots with applications to Gaussian processes and Bayesian optimization*, in Adv. Neural
          Inf. Process. Syst., vol. 33, 2020, pp. 22268–22281.

[41] W. Rudin, *Real and Complex Analysis*, McGraw-Hill, New York, NY, 3 ed., 1987.

[42] Y. Saad, *Analysis of some Krylov subspace approximations to the matrix exponential operator*,
          SIAM J. Numer. Anal., 29 (1992), pp. 209–228, https://doi.org/10.1137/0729014.

[43] R. L. Schilling, R. Song, and Z. Vondracek, *Bernstein Functions*, De Gruyter, Berlin,
          Germany, 2012, https://doi.org/10.1515/9783110269338.

[44] 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, https://doi.org/10.1016/j.laa.2016.06.041.

[45] M. Schweitzer, *Restarting and error estimation in polynomial and extended Krylov subspace
          methods for the approximation of matrix functions*, PhD thesis, Bergische Universität
          Wuppertal, Fakultät für Mathematik und Naturwissenschaften, 2016.

[46] L. F. Shampine, *Vectorized adaptive quadrature in MATLAB*, J. Comput. Appl. Math., 211
          (2008), pp. 131–140, https://doi.org/10.1016/j.cam.2006.11.021.

[47] G. E. Shilov, *Elementary Real and Complex Analysis*, MIT Press, Cambridge, 1973.

[48] 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, https://doi.org/10.1137/040617868.

[49] 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, https://doi.
          org/10.1137/16M1063733.

[50] D. V. Widder, *The Laplace Transform*, Princeton Mathematical Series, Princeton University
          Press, Princeton, NJ, 1946.

[51] F. J. Wilson, *The representation of functions as Laplace and Laplace–Stieltjes transforms*,
          SIAM J. Math. Anal., 17 (1986), pp. 720–733, https://doi.org/10.1137/0517051.