Bergische Universität Wuppertal

Fachbereich Mathematik und Naturwissenschaften

Institute of Mathematical Modelling, Analysis and Computational Mathematics (IMACM)

Preprint BUW-IMACM 15/21

Andreas Frommer, Marcel Schweitzer

Error bounds and estimates for Krylov subspace approximations of Stieltjes matrix functions

April 27, 2015

http://www.math.uni-wuppertal.de

Noname manuscript No.
(will be inserted by the editor)

Error bounds and estimates for Krylov subspace approximations of Stieltjes matrix functions

Andreas Frommer · Marcel Schweitzer

Received: date / Accepted: date

Abstract When using the Lanczos method to approximate $f(A)\boldsymbol{b}$, the action of a matrix function on a vector, there is, in contrast to the solution of linear systems, no straightforward way to measure or estimate the error of the current iterate. Therefore, to be able to decide whether the desired accuracy has been reached, several different estimates and bounds for the error have been suggested, all of them specific to certain classes of functions. In this paper, we add to these results by developing a technique to compute error bounds for Stieltjes functions, using a recently suggested integral representation of the error, and we show how these bounds can be computed essentially for free, i.e., with cost independent of the iteration number and the dimension of the matrix A.

Keywords matrix function · Stieltjes function · Krylov subspace approximation · Lanczos method · error estimates · Gaussian quadrature

Mathematics Subject Classification (2000) 65F60, 65F50, 65F10, 65F30, 65D30, 65D32

1 Introduction

Given a Hermitian positive definite matrix $A \in \mathbb{C}^{n\times n}$, a vector $\boldsymbol{b} \in \mathbb{C}^n$ and a sufficiently smooth function f, a common and important task in numerical simulations is the computation of $f(A)\boldsymbol{b}$, the action of the matrix function $f(A)$ on the vector $\boldsymbol{b}$. Important examples are the exponential function $f(z) = e^z$ used in exponential integrators for the solution of differential equations [30,31], the matrix sign function in lattice quantum chromodynamics [6,42] and fractional powers $f(z) = z^\alpha$ in fractional differential equations [7,36] or sampling from Gaussian-Markov random fields [32].

The direct computation of $f(A)$ is often not feasible as the matrix A is typically large and sparse in most applications, e.g., a discretization of a differential operator,

This work was supported by Deutsche Forschungsgemeinschaft through Cooperative Research Centre SFB TRR55 “Hadron Physics from Lattice QCD”.

Department of Mathematics, Bergische Universität Wuppertal, 42097 Wuppertal, Germany,
{frommer,schweitzer}@math.uni-wuppertal.de

but $f(A)$ is a dense matrix in general; cf., e.g., [29]. Therefore it is common practice to approximate $f(A)\boldsymbol{b}$ directly by some iterative method. The most prominent class of iterative methods for matrix functions are *Krylov subspace methods* [10, 22, 30, 33, 37, 43] and among these *Arnoldi's method* which reduces to the *Lanczos method* in the Hermitian case. In the mth step of this method, an orthonormal basis $V = [\boldsymbol{v}_1 \mid \cdots \mid \boldsymbol{v}_m]$ of the mth Krylov subspace $\mathcal{K}_m(A, \boldsymbol{b}) = \operatorname{span}\{\boldsymbol{b}, A\boldsymbol{b}, \ldots, A^{m-1}\boldsymbol{b}\}$ is constructed which fulfills the *Lanczos relation*

$$AV_m = V_m T_m + t_{m+1,m}\boldsymbol{v}_{m+1}\boldsymbol{e}_m^T,$$

where $T_m \in \mathbb{C}^{m\times m}$ is a tridiagonal matrix and $\boldsymbol{e}_m$ denotes the mth canonical unit vector. Given this decomposition, the mth Lanczos approximation for $f(A)\boldsymbol{b}$ is computed as

$$\boldsymbol{f}_m = V_m f(T_m) V_m^H \boldsymbol{b} = \|\boldsymbol{b}\| V_m f(T_m)\boldsymbol{e}_1, \tag{1}$$

where, as everywhere in this paper unless stated otherwise, $\|\cdot\|$ denotes the Euclidean vector norm.

In the special case $f(z) = z^{-1}$ where the computation of $f(A)\boldsymbol{b}$ corresponds to solving the linear system $A\boldsymbol{x} = \boldsymbol{b}$, the iterates produced by the Lanczos method are identical to those of the *conjugate gradient method* for linear systems [28].

One major difference when computing $f(A)\boldsymbol{b}$ for general f instead of solving a linear system is that no straight-forward stopping criterion is available. When solving a linear system, one can easily and efficiently compute the norm of the residual $\boldsymbol{r}_m = \boldsymbol{b} - A\boldsymbol{f}_m$ to monitor the progress of the method. Using the relation

$$A\boldsymbol{e}_m = \boldsymbol{r}_m$$

for the error $\boldsymbol{e}_m = A^{-1}\boldsymbol{b} - \boldsymbol{f}_m$ one can measure the error via $\|\boldsymbol{r}\|$, the A^HA-energy norm of $\boldsymbol{e}$, or obtain the simple error bound

$$\|A^{-1}\boldsymbol{b} - \boldsymbol{f}_m\| \leq \frac{1}{\lambda_{\min}}\|\boldsymbol{r}_m\|,$$

where $\lambda_{\min}$ denotes the smallest eigenvalue of A, which is supposed to be known. One can also use more sophisticated techniques to compute error bounds, e.g., based on Gauss quadrature, see [25].

As one does not have a residual available for general f, one has to resort to other techniques for deciding at which point the Lanczos iteration can be terminated because the current iterate fulfills the desired accuracy requirements.

Due to this fact, various estimates and bounds for $\|f(A)\boldsymbol{b} - \boldsymbol{f}_m\|$ have been considered in the literature, e.g., for the exponential function [30], for the sign function [42] or for rational functions [18, 20, 21]. All these bounds and estimates have in common that they only apply to special functions or classes of functions and do not hold for arbitrary f. In this paper, we add to these results by considering another class of functions, the *Stieltjes functions*, see, e.g. [4, 5, 27], i.e., functions f which can be defined via a Riemann–Stieltjes integral

$$f(z) = \int_0^\infty \frac{1}{t+z}\,\mathrm{d}\mu(t), \quad z \in \mathbb{C} \setminus (-\infty, 0] \tag{2}$$

with respect to some positive, monotone function $\mu(t)$, also called the *generating function* of f. Two important examples of Stieltjes functions relevant in practical applications are

$$z^{-\alpha} = \frac{\sin(\alpha\pi)}{\pi} \int_0^\infty \frac{t^{-\alpha}}{t+z} \,\mathrm{d}t \quad \text{for } \alpha \in (0,1)$$

and

$$\frac{\log(1+z)}{z} = \int_1^\infty \frac{t^{-1}}{t+z} \,\mathrm{d}t.$$

For more examples of Stieltjes functions see, e.g., [4, 5, 27].

Using a result from [15,16], one can see that the error of the Lanczos approximation $\boldsymbol{f}_m$ for a Stieltjes function of the form (2) is given as the action of a Stieltjes function corresponding to a different generating function $\widetilde{\mu}$ applied to the next Lanczos basis vector $\boldsymbol{v}_{m+1}$. This representation can be used to characterize the norm of the error as a bilinear form which is then bounded using the theory of Golub and Meurant relating moments of the form $\boldsymbol{v}^H g(A)\boldsymbol{v}$ to Gaussian quadrature [24–26]. This is very similar to what was presented in [18] for certain rational functions in partial fraction form. These are a subset of the class of Stieltjes functions and in fact, both approaches lead to the same results in this case. The approach of [18] is applicable to a broader class of functions in the sense that one can first look for a suitable rational approximation $r \approx f$ and then work with r. Our approach, however, works with f directly and does not require the a priori knowledge of a rational approximation, which may be difficult or costly to construct. In addition, if one is interested in bounds for the error with respect to $f(A)\boldsymbol{b}$ (and not $r(A)\boldsymbol{b}$), one needs to take the sign of the error in the rational approximation into account, which severely limits the situation in which it can be guaranteed that the computed estimates are guaranteed bounds. This problem can also be circumvented by working with f directly.

The remainder of this paper is organized as follows. In Section 2 we briefly review the Lanczos process and the theory of Golub and Meurant relating it to Gaussian quadrature and how it can be used to bound or estimate certain bilinear forms. In Section 3, we recall the integral representation of the error for Stieltjes functions given in [15, 16] and use it to derive methods for computing error bounds for Lanczos approximations. In Section 4 we give a detailed description of the Lanczos algorithm with computation of error bounds, and we focus on how to perform all necessary additional computations with cost independent of the size n of the matrix as well as of the iteration number m. We briefly comment on extensions of the developed techniques to the non-Hermitian case in Section 5, to obtain error estimates (not bounds in general) for the error of the Arnoldi approximation, albeit with computational cost now depending on m. In Section 6, we investigate various numerical test cases from applications (involving both Hermitian and non-Hermitian matrices), both to demonstrate the quality of our proposed error bounds and estimates and to study the dependence of our method on different parameters. Concluding remarks are given in Section 7.

2 The Lanczos process and its relation to Gaussian quadrature

The Lanczos process is the simplified version of Arnoldi's method for Hermitian matrices. It is based on the well-known short recurrence relation for the orthonormal Krylov basis vectors that can be employed in this case and is summarized in Algorithm 1;

cf. , e.g., [34, 37]. Once the matrices V_m and T_m are computed by Algorithm 1, an approximation for $f(A)\boldsymbol{b}$ can be computed via (1).

Algorithm 1: Lanczos process for building an orthonormal basis of $\mathcal{K}_{m_{\max}}(A, \boldsymbol{b})$

```
Given: $A$, $\boldsymbol{b}$, $m$
Set $\boldsymbol{v}_0 := \boldsymbol{0}$, $\boldsymbol{v}_1 := \boldsymbol{b}/\|\boldsymbol{b}\|$ and $t_{1,0} := 0$.
for $m = 1, 2, \ldots, m_{\max}$ do
    Set $\boldsymbol{w}_m := A\boldsymbol{v}_k - t_{m,m-1}\boldsymbol{v}_{m-1}$.
    Set $t_{m,m} := (\boldsymbol{w}_m, \boldsymbol{v}_m)$.
    Set $\boldsymbol{w}_m := \boldsymbol{w}_m - t_{m,m}\boldsymbol{v}_m$.
    Set $t_{m+1,m} := \|\boldsymbol{w}_m\|$.
    if $t_{m+1,m} = 0$ then
        Stop.
    Set $\boldsymbol{v}_{m+1} := \boldsymbol{w}_j / t_{m+1,m}$.
```

In the context of this paper, the Lanczos algorithm is of utmost importance not only because it builds the basis for computing the Lanczos approximation (1) for $f(A)\boldsymbol{b}$, but also because it can be related to orthogonal polynomials and thus Gaussian quadrature as pointed out in [24–26]: The *Lanczos polynomials*, i.e., the polynomials p_{k-1} of degree $k-1$ with $p_{k-1}(A)\boldsymbol{v}_1 = \boldsymbol{v}_k$, are orthogonal with respect to the inner product

$$\langle p, q \rangle_\alpha = \int_a^b p(t)q(t)\,\mathrm{d}\alpha(t), \tag{3}$$

where $\alpha(t)$ is a step function depending on the eigenvalues and eigenvectors of A and $\operatorname{spec}(A) \subseteq [a, b]$. More precisely, if $\hat\lambda_1 < \hat\lambda_2 < \cdots < \hat\lambda_{\hat n}$ denote the *distinct* eigenvalues of A with corresponding eigenvectors $\hat{\boldsymbol{v}}_i$, α is given by

$$\alpha(t) = \begin{cases} 0, & \text{if } t < \hat\lambda_1, \\ \sum_{j=1}^{i} (\eta_j)^2, & \text{if } \hat\lambda_i \le t < \hat\lambda_{i+1}, \\ \sum_{j=1}^{\hat n} (\eta_j)^2, & \text{if } \hat\lambda_{\hat n} < t, \end{cases} \tag{4}$$

where the coefficients η_j stem from the decomposition $\boldsymbol{v}_1 = \sum_{j=1}^{\hat n} \eta_j \hat{\boldsymbol{v}}_j$; see, e.g., [26]. For ease of notation, we will from now on always assume that A has pairwise distinct eigenvalues $\lambda_1 < \lambda_2 < \cdots < \lambda_n$. Using (3) and (4), one can show the following theorem, relating Gaussian quadrature to matrix functions of T_m.

Theorem 1 (Theorem 4.2 and 6.6 in [26]) *Let $A \in \mathbb{C}^{n\times n}$ be Hermitian positive definite with smallest and largest eigenvalue λ_1 and λ_n respectively, $\boldsymbol{v}_1 \in \mathbb{C}^n$ with $\|\boldsymbol{v}_1\| = 1$, let h be a function defined on $[\lambda_1, \lambda_n]$ and let α be defined as in (4). Let $t_\ell, \omega_\ell, \ell = 1, \ldots, m$ be the nodes and weights of the m-point Gaussian quadrature rule for approximating $\int_{\lambda_1}^{\lambda_n} h(t)\,\mathrm{d}\alpha(t)$. Then*

$$\sum\nolimits_{\ell=1}^{m} \omega_\ell h(t_\ell) = \boldsymbol{e}_1^H h(T_m)\boldsymbol{e}_1,$$

where T_m is the tridiagonal matrix obtained by m steps of the Lanczos algorithm applied to A and $\boldsymbol{v}_1$.

In the same way, the $(m+1)$-point Gauss–Radau quadrature rule (a quadrature rule in which one node is fixed to be at the left endpoint λ_1 of the integration interval) can be evaluated as $\boldsymbol{e}_1^H h(\widetilde{T}_{m+1})\boldsymbol{e}_1$, with the modified tridiagonal matrix

$$\widetilde{T}_{m+1} = \begin{bmatrix} T_m & \beta_m \boldsymbol{e}_m \\ \beta_m \boldsymbol{e}_m^H & \boldsymbol{d}_m \end{bmatrix}, \text{ where } \boldsymbol{d} = \beta_m^2 (T_m - \lambda_1 I)^{-1} \boldsymbol{e}_m; \tag{5}$$

see [26]. One important feature of Gauss and Gauss–Radau quadrature is that they give lower and upper bounds for $\int_{\lambda_1}^{\lambda_n} h(t)\,\mathrm{d}\alpha(t)$ if the derivatives of h have constant sign on $[\lambda_1, \lambda_n]$; see, e.g., [9, 26].

Theorem 2 *Let h be $(2m+1)$-times continuously differentiable on $[\lambda_1, \lambda_n]$. Let t_ℓ, ω_ℓ and $\widetilde{t}_\ell, \widetilde{\omega_\ell}$ be the nodes and weights of the m-point Gauss and $(m+1)$-point Gauss-Radau quadrature rule (with one node fixed at λ_1) for $\int_{\lambda_1}^{\lambda_n} h(t)\,\mathrm{d}\alpha(t)$, respectively. If $h^{(2m)}(t) \geq 0$ for $t \in [\lambda_1, \lambda_n]$, then*

$$\sum\nolimits_{\ell=1}^{m} \omega_\ell h(t_\ell) \leq \int_{\lambda_1}^{\lambda_n} h(t)\,\mathrm{d}\alpha(t).$$

If $h^{(2m+1)}(t) \leq 0$ for $t \in [\lambda_1, \lambda_n]$, then

$$\sum\nolimits_{\ell=1}^{m+1} \widetilde{\omega}_\ell h(\widetilde{t}_\ell) \geq \int_{\lambda_1}^{\lambda_n} h(t)\,\mathrm{d}\alpha(t).$$

According to Theorem 2, Gauss and Gauss–Radau quadrature always give lower and upper bounds for $\int_{\lambda_1}^{\lambda_n} h(t)\,\mathrm{d}\alpha(t)$, respectively, if

$$(-1)^{m+1} h^{(m)}(t) \geq 0 \text{ for all } m, \text{ and } t \in [\lambda_1, \lambda_n]. \tag{6}$$

Functions with the property (6) are called *completely monotonic on* $[\lambda_1, \lambda_n]$. One easily sees that every Stieltjes function is completely monotonic on any interval $[\lambda_1, \lambda_n] \subseteq (0, \infty)$. We summarize this and another important property in the following proposition; see, e.g., [2, 4].

Proposition 1 *i) Let f be a Stieltjes function. Then f is completely monotonic on any interval $[\lambda_1, \lambda_n] \subseteq (0, \infty)$.*
ii) Let f, g be completely monotonic on $[\lambda_1, \lambda_n]$. Then $f \cdot g$ is completely monotonic on $[\lambda_1, \lambda_n]$.

In the light of Proposition 1, to be able to compute upper and lower bounds (and not just estimates) for the norm of the error of the Lanczos approximation to $f(A)\boldsymbol{b}$ by Gaussian quadrature, it suffices to show that it can be expressed as a Riemann–Stieltjes integral of a product of Stieltjes (or other completely monotonic) functions. A representation of this type will be derived in the next section.

3 Representations and bounds for the Lanczos error

In [12, 32, 41], a representation of the error in terms of divided differences was given. This representation turns out to be highly unstable in finite precision arithmetic in contrast to the alternative integral representation of the error first given in [16], and rephrased for Stieltjes functions in [15]. This representation is the key to obtaining our error bounds.

Theorem 3 (Theorem 2.1 in [15]) *Let f be a Stieltjes function, let $A \in \mathbb{C}^{n\times n}$ be Hermitian positive definite, $\boldsymbol{b} \in \mathbb{C}^n$ and denote by $\boldsymbol{f}_m$ the mth Lanczos approximation* (1) *to $f(A)\boldsymbol{b}$. Define*

$$e_m(z) = (-1)^{m+1}\|\boldsymbol{b}\|\gamma_m \int_0^\infty \frac{1}{w_m(t)} \cdot \frac{1}{z+t}\,\mathrm{d}\mu(t), \quad z \in \mathbb{C}\setminus(-\infty,0], \tag{7}$$

where $w_m(t) = (t+\theta_1)\cdots(t+\theta_m)$ with $\operatorname{spec}(T_m) = \{\theta_1,\ldots,\theta_m\}$ and $\gamma_m = \prod_{i=1}^m t_{i+1,i}$. Then

$$f(A)\boldsymbol{b} - \boldsymbol{f}_m = e_m(A)\boldsymbol{v}_{m+1}, \tag{8}$$

where $\boldsymbol{v}_{m+1}$ is the $(m+1)$st Lanczos vector.

In [15] it is shown that the error function $e_m(z)$ from (7) is again a (scalar multiple of a) Stieltjes function if A is Hermitian positive definite. To be specific, $e_m(z) = (-1)^{m+1}\|\boldsymbol{b}\|\gamma_m\widetilde{e}_m(z)$ where

$$\widetilde{e}_m(z) = \int_0^\infty \frac{1}{t+z}\,\mathrm{d}\widetilde{\mu}(t) \text{ with } \mathrm{d}\widetilde{\mu}(t) = \frac{1}{w_m(t)}\,\mathrm{d}\mu(t). \tag{9}$$

Using Theorem 3, we straight-forwardly find a representation for the norm of the Lanczos error as a bilinear form.

Corollary 1 *Let the assumptions of Theorem 3 hold. Then*

$$\|f(A)\boldsymbol{b} - \boldsymbol{f}_m\|^2 = \|\boldsymbol{b}\|^2\gamma_m^2\boldsymbol{v}_{m+1}^H\widetilde{e}_m(A)^2\boldsymbol{v}_{m+1}, \tag{10}$$

where $\widetilde{e}_m(z)$ is given by (9).

Proof We have

$$\begin{aligned}\|f(A)\boldsymbol{b} - \boldsymbol{f}_m\|^2 &= \left((-1)^{m+1}\|\boldsymbol{b}\|\gamma_m\widetilde{e}_m(A)\boldsymbol{v}_{m+1}\right)^H\left((-1)^{m+1}\|\boldsymbol{b}\|\gamma_m\widetilde{e}_m(A)\boldsymbol{v}_{m+1}\right)\\ &= \|\boldsymbol{b}\|^2\gamma_m^2\boldsymbol{v}_{m+1}^H\left(\widetilde{e}_m(A)\right)^H\widetilde{e}_m(A)\boldsymbol{v}_{m+1}\\ &= \|\boldsymbol{b}\|^2\gamma_m^2\boldsymbol{v}_{m+1}^H\widetilde{e}_m(A)^2\boldsymbol{v}_{m+1},\end{aligned}$$

where the last equality holds because $\widetilde{e}_m(A)$ is Hermitian if A is Hermitian.

Using the spectral decomposition $A = Q\Lambda Q^H$ of A and the short-hand notation $\boldsymbol{u} = Q\boldsymbol{v}_{m+1}$ one can rewrite the bilinear form from the right-hand side of (10) as

$$\boldsymbol{v}_{m+1}\widetilde{e}_m(A)^2\boldsymbol{v}_{m+1} = \boldsymbol{u}^H\widetilde{e}_m(\Lambda)^2\boldsymbol{u} = \sum\nolimits_{i=1}^n \widetilde{e}_m(\lambda_i)^2\boldsymbol{u}_i^2 = \int_{\lambda_1}^{\lambda_n}\widetilde{e}(t)^2\,\mathrm{d}\alpha(t), \tag{11}$$

with

$$\alpha(t) = \begin{cases} 0, & \text{if } t < \lambda_1, \\ \sum_{j=1}^{i} (\boldsymbol{u}_j)^2, & \text{if } \lambda_i \leq t < \lambda_{i+1}, \\ \sum_{j=1}^{n} (\boldsymbol{u}_j)^2, & \text{if } \lambda_n < t; \end{cases} \tag{12}$$

see also [26, Chapter 7]. Comparing (12) and (4), one sees that the function α is exactly the one generated by the Lanczos process started with A and $\boldsymbol{v}_{m+1}$. Therefore, according to Theorem 1, the integral on the right-hand side of (11) can be approximated by m-point Gaussian quadrature as follows: First compute T_m as a result of m steps of the Lanczos process for A and $\boldsymbol{v}_{m+1}$ and then evaluate $\boldsymbol{e}_1^H \widetilde{(T_m)^2} \boldsymbol{e}_1$. Thus, it is possible to evaluate the Riemann–Stieltjes integral without explicitly knowing the measure α (and thus the eigenvalues of A). As $\widetilde{e}(t)^2$ is, by (9), the product of two Stieltjes functions, it is completely monotonic by Proposition 1 and we therefore can conclude from Theorem 2 that this procedure will give a lower bound for the exact value of the integral. Of course, all of the above applies in an analogous way for Gauss–Radau quadrature. We summarize these findings in the following theorem.

Theorem 4 *Let f be a Stieltjes function, let $A \in \mathbb{C}^{n\times n}$ be Hermitian positive definite, $\boldsymbol{b} \in \mathbb{C}^n$ and denote by $\boldsymbol{f}_m$ the mth Lanczos approximation (1) to $f(A)\boldsymbol{b}$. Let $\boldsymbol{v}_{m+1}$ be the $(m+1)$st Lanczos vector and $\gamma_m = \prod_{i=1}^{m} t_{i+1,i}$. Denote by $T_k^{(2)}$ the tridiagonal matrix resulting from k steps of the Lanczos process applied to A and $\boldsymbol{v}_{m+1}$ and by $\widetilde{T}_{k+1}^{(2)}$ the modification of $T_k^{(2)}$ according to (5). Then*

$$\|\boldsymbol{b}\|^2 \gamma_m^2 \boldsymbol{e}_1^H \widetilde{e}\left(T_k^{(2)}\right)^2 \boldsymbol{e}_1 \leq \|f(A)\boldsymbol{b} - \boldsymbol{f}_m\|^2 \leq \|\boldsymbol{b}\|^2 \gamma_m^2 \boldsymbol{e}_1^H \widetilde{e}\left(\widetilde{T}_{k+1}^{(2)}\right)^2 \boldsymbol{e}_1. \tag{13}$$

Theorem 4 gives a way of computing bounds for the error of the Lanczos approximation to $f(A)\boldsymbol{b}$. However, two issues seemingly prevent the result of Theorem 4 from being useful for practical applications. First, the error function $\widetilde{e}$ is in general not known explicitly and can not be evaluated exactly; see also [16]. Second, computing the error bounds (13) by a k-point Gaussian quadrature rule (or a $(k+1)$-point Gauss–Radau rule) in the way suggested by Theorem 4 would require k additional matrix vector multiplications with A which do not contribute to advancing the Lanczos iteration for approximating $f(A)\boldsymbol{b}$. Investing this much work for computing the error bounds is in general not feasible in practical applications. The first issue is covered in the remainder of this section while avoiding the additional multiplications with A is one of the topics of Section 4.

In [16], the error representation (8) was used for restarting the Arnoldi iteration for the computation of $f(A)\boldsymbol{b}$. There, an “inner” numerical quadrature rule was used to evaluate the not explicitly known function $\widetilde{e}(z)$. Of course, this is also possible for computing the matrix functions in (13), leading to error estimates computed by two nested quadrature rules. For these estimates to be *bounds* again, we therefore have to take care of computing lower (resp. upper) bounds in the inner quadrature as well. The proper choice of the inner quadrature rule for this purpose largely depends on the function to be approximated and on how we take care of the infinite interval of integration. Assuming that we have a quadrature rule at hand that gives bounds for (8) in the scalar case, the following result guarantees that it will also compute bounds for the bilinear forms on the left and right-hand side of (13).

Proposition 2 *Let the assumptions of Theorem 3 hold and let T be any symmetric positive definite matrix. Further, let $t_\ell, \omega_\ell, \ell = 1, \ldots, k$ be the nodes and weights of a quadrature rule which gives lower bounds, i.e., for which*

$$\sum\nolimits_{\ell=1}^{k} \frac{\omega_\ell}{t_\ell + z} \le \widetilde{e}_m(z) \text{ for } z \in (0,\infty). \tag{14}$$

Then

$$\boldsymbol{e}_1^H \left(\sum\nolimits_{\ell=1}^{k} \omega_\ell (t_\ell I + T)^{-1}\right)^2 \boldsymbol{e}_1 \le \boldsymbol{e}_1^H \widetilde{e}_m(T)^2 \boldsymbol{e}_1.$$

The result holds analogously for quadrature rules which give upper bounds. In particular, the result applies to the matrices $T_k^{(2)}$ and $\widetilde{T}_k^{(2)}$ from Theorem 4.

Proof Using the spectral decomposition $T = UDU^H$ with diagonal matrix D and defining the short-hand notation $\boldsymbol{u} = U^H \boldsymbol{e}_1$, we have

$$\begin{aligned} \boldsymbol{e}_1^H \left(\sum\nolimits_{\ell=1}^{k} \omega_\ell (t_\ell I + T)^{-1}\right)^2 \boldsymbol{e}_1 &= \boldsymbol{u}^H \left(\sum\nolimits_{\ell=1}^{k} \omega_\ell (t_\ell I + D)^{-1}\right)^2 \boldsymbol{u} \\ &= \sum\nolimits_{i=1}^{m} |\boldsymbol{u}_i|^2 \left(\sum\nolimits_{\ell=1}^{k} \frac{\omega_\ell}{t_\ell + d_{ii}}\right)^2. \end{aligned} \tag{15}$$

Using (14), we can bound the right-hand side of (15) by

$$\sum\nolimits_{i=1}^{m} |\boldsymbol{u}_i|^2 \left(\sum\nolimits_{\ell=1}^{k} \frac{\omega_\ell}{t_\ell + d_{ii}}\right)^2 \le \sum\nolimits_{i=1}^{m} |\boldsymbol{u}_i|^2 \widetilde{e}_m(d_{ii})^2 = \boldsymbol{u}_1^H \widetilde{e}_m(D)^2 \boldsymbol{u}_1 = \boldsymbol{e}_1^H \widetilde{e}_m(T) \boldsymbol{e}_1,$$

which concludes the proof for lower bounds. The modifications necessary for proving the result for upper bounds are straight-forward. The matrix $T_k^{(2)}$ is obviously Hermitian positive definite, as A is Hermitian positive definite and $T_k^{(2)} = V^H AV$ for a matrix V of full rank. For $\widetilde{T}_{k+1}^{(2)}$ note that the modification (5) again results in a Hermitian matrix. Its eigenvalues are the nodes of a Gauss–Radau rule, which are known to lie in the interval $[\lambda_{\min}, \lambda_{\max}]$ of integration, see, e.g., [23], so that $\widetilde{T}_{k+1}^{(2)}$ is also Hermitian positive definite.

Let us note, however, that our experiments reported in Section 6 suggest that in practical applications, the error in the *outer* Gaussian quadrature rules (13) is typically significantly larger than the error of the inner quadrature rule for practically feasible choices of parameters. Therefore one can expect the computed quantities to be bounds even if the inner quadrature rule does not guarantee this property, such that the above considerations are mainly of theoretical interest.

4 Computing error bounds with low computational cost

Computing error bounds by Gaussian quadrature with k nodes for the Lanczos iterate $\boldsymbol{f}_m$ directly in the way suggested by Theorem 4 requires k additional matrix vector multiplications with A (which do not serve to advance the original Lanczos process) as the tridiagonal matrix $T_k^{(2)}$ resulting from the Lanczos process applied to A and $\boldsymbol{v}_{m+1}$ is needed. However, a result from [18] shows that this matrix can be computed without investing any additional matrix vector multiplications with A.

Theorem 5 (Theorem 4.1 in [18]) *Let T_{m+1+k} be the tridiagonal matrix resulting from $k+m+1$ steps of the Lanczos process for A and $\boldsymbol{v}_1$. Let $\hat{k} = \min\{m, k\}$ and denote by $\widetilde{T}$ the lower right $(k + \hat{k} + 1) \times (k + \hat{k} + 1)$ sub-block of T_{m+1+k} and let $\hat{T}$ denote the tridiagonal matrix resulting from k steps of the Lanczos process applied to $\widetilde{T}$ and $\mathbf{e}_{\hat{k}+1}$. Then $\hat{T} = T_k^{(2)}$, where $T_k^{(2)}$ denotes the matrix resulting from k iterations of the Lanczos process for A and $\boldsymbol{v}_{m+1}$.*

By Theorem 5, the matrix $T_k^{(2)}$ can be computed by applying k steps of the Lanczos process to a tridiagonal matrix of size of size $2k + 1$ (or less if $m < k$), i.e., with computational cost $\mathcal{O}(k^2)$, independent of the size of the original matrix A. The price for this reduction in computational cost is that the matrix needed for computing bounds for the error of the mth Lanczos iterate can only be computed *in retrospect* in the $(m+k+1)st$ step, i.e., the more quadrature nodes are used (and thus the more accurate the error bound is), the later the bound is available. We will further comment on and investigate this trade-off between accuracy and timely availability of the error bounds in the numerical experiments reported in Section 6.

To be able to compute error bounds according to Theorem 4, one needs, in addition to the matrix $T_k^{(2)}$, to evaluate the nodal polynomial $w_m(t)$ at the quadrature nodes, to be able to approximately evaluate the error function $\widetilde{e}$. The polynomial $w_m(t)$ is of degree m, so a naive approach of evaluating it at ℓ quadrature nodes $t_i, i = 1, \dots, \ell$ would require at least $\mathcal{O}(m\ell)$ arithmetic operations and the computational cost would thus not be independent of the iteration number (albeit growing only linearly in m and thus still being negligible in practice as long as m stays small in comparison to n). Fortunately, one can easily compute $w_m(t_i)$ with cost $\mathcal{O}(1)$ if $w_{m-1}(t_i)$ is known by using simple recursion relations, see lines 14–16 of Algorithm 2, where $\rho_i(m)$ stores the value $w_m(t_i)$. These relations can, e.g., be derived from inspecting Gaussian elimination for the matrix $T_m + tI$ or from certain versions of the (shifted) CG algorithm [18, 19, 38], where they arise naturally. Using these update formulas requires fixing the quadrature nodes for the inner quadrature rule in advance and using the same nodes throughout all iterations. This is, however, no problem in practice: adaptively changing the quadrature nodes to reach higher accuracy is in general not necessary, as we are only interested in computing bounds which do not necessitate $\widetilde{e}$ to be approximated up to machine precision. In particular, the number k of quadrature nodes in the outer Gaussian quadrature will in practice be much smaller than ℓ, such that the quadrature error of the inner rule will not make the main contribution to the sharpness of the resulting bounds; see also Section 6 for a discussion of numerical results on this topic.

Algorithm 2 summarizes how to incorporate the efficient computation of retrospective error bounds into the Lanczos process for $f(A)\boldsymbol{b}$. The upper bound for the error norm is used as a stopping criterion for the iteration in Algorithm 2. In case that the upper bound in iteration m lies below the specified tolerance `tol`, we form the Arnoldi approximation $\boldsymbol{f}_m$ and return it as result of the algorithm, although the error bound computed corresponds to the approximation $\boldsymbol{f}_{m-k}$. By a result from [14], it is known that the Euclidean norm of the error of the Lanczos approximation to $f(A)\boldsymbol{b}$ for A Hermitian and f a Stieltjes function is monotonically decreasing, so that the upper bound for the error of $\boldsymbol{f}_{m-k}$ is also valid for $\boldsymbol{f}_m$ and we thus have the guarantee that the returned approximation fulfills the accuracy requirement.

The next result summarizes the additional computational cost of Algorithm 2 in comparison to Algorithm 1.

Algorithm 2: Lanczos approximation with error bounds for $f(A)\boldsymbol{b}$

```
   Given: $A$, $\boldsymbol{b}$, $f$, $k$, $\ell$, tol
1  Choose quadrature nodes/weights $(t_i, \omega_i)_{i=1,\ldots,\ell}$.   {inner quadrature rule}
2  Initialize $d_i(0) \leftarrow 1, \rho_i(0) \leftarrow 1, i = 1, \ldots, \ell$.
3  Initialize $t_{1,0} \leftarrow 0$.
4  $\boldsymbol{v}_1 \leftarrow \frac{1}{\|\boldsymbol{b}\|}\boldsymbol{b}$
5  for $m = 1, 2, \ldots$ do
6      Compute $\boldsymbol{w}_m \leftarrow A\boldsymbol{v}_m - h_{m,m-1}\boldsymbol{v}_{m-1}$
7      $t_{m,m} \leftarrow (\boldsymbol{w}_m, \boldsymbol{v}_m)$
8      $\boldsymbol{w}_m \leftarrow \boldsymbol{w}_m - t_{m,m}\boldsymbol{v}_m$
9      $t_{m+1,m} \leftarrow \|\boldsymbol{w}_m\|$
10     if $t_{m+1,m} = 0$ then
11         compute $\boldsymbol{f}_m \leftarrow \|\boldsymbol{b}\|V_m f(T_m)\boldsymbol{e}_1$.
12         Stop.
13     $\boldsymbol{v}_{m+1} \leftarrow \frac{1}{t_{m+1,m}}\boldsymbol{w}_m$
14     for $i = 1, \ldots, \ell$ do
15         $d_i(m) \leftarrow (t_{m,m} + t_i) - \frac{t_{m+1,m}^2}{d_i(m-1)}$
16         $\rho_i(m) \leftarrow \rho_i(m-1) \cdot \frac{t_{m+1,m}}{d_i(m)}$
17     Set $\hat{k} \leftarrow \min\{m+1, k+1\}$.
18     Let $\widetilde{T}$ be the lower right $(k+\hat{k}) \times (k+\hat{k})$ sub-block of $T_m$.
19     Perform $k$ steps of Algorithm 1 for $\widetilde{T}$ and $\boldsymbol{e}_{\hat{k}}$, yielding $\hat{T}$.
20     Modify $\hat{T}$ according to (5), yielding $\bar{T}$.
21     Compute lower_bound $\leftarrow \|\boldsymbol{b}\|^2\boldsymbol{e}_1^H\left(\sum_{i=1}^{\ell}\omega_i\rho_i(m-k)(\hat{T}+t_iI)^{-1}\right)^2\boldsymbol{e}_1$
22     Compute upper_bound $\leftarrow \|\boldsymbol{b}\|^2\boldsymbol{e}_1^H\left(\sum_{i=1}^{\ell}\omega_i\rho_i(m-k)(\bar{T}+t_iI)^{-1}\right)^2\boldsymbol{e}_1$
23     if upper_bound $\leq$ tol then
24         Compute $\boldsymbol{f}_m \leftarrow \|\boldsymbol{b}\|V_m f(T_m)\boldsymbol{e}_1$.
25         Stop.
```

Lemma 1 *Performing Algorithm 2 instead of Algorithm 1 (plus the computation of $\boldsymbol{f}_m$) for $A \in \mathbb{C}^{n\times n}$ and $\boldsymbol{b} \in \mathbb{C}^n$ requires an additional computational cost of the order $\mathcal{O}(k^2 + k\ell)$ per iteration and thus an overall additional work of $\mathcal{O}(m_{\max}k^2 + m_{\max}k\ell)$, if $m_{\max}$ iterations are necessary to reach the desired accuracy. In particular, the additional cost in the mth iteration is independent of both, m and n.*

Proof The initializations in line 1–3 of Algorithm 2 have cost $\mathcal{O}(\ell)$, assuming that the nodes and weights of the quadrature rule are available and do not need to be computed by a separate algorithm. Line 4–13 (ignoring the computation of $\boldsymbol{f}_m$ in line 11) exactly correspond to the Lanczos process given in Algorithm 1. The **for** loop in line 14–16 has computational cost $\mathcal{O}(\ell)$, as the update formulas for d_i and ρ_i only require a fixed number of scalar operations. Line 17 has cost $\mathcal{O}(1)$. Line 18 has cost $\mathcal{O}(k)$ and line 19 has cost $\mathcal{O}(k^2)$, as $\widetilde{T}$ is tridiagonal and matrix vector products with it can therefore be formed with cost $\mathcal{O}(k)$. Line 20 again has cost $\mathcal{O}(k)$ for solving the linear system with $\widetilde{T}$. The computation of the lower and upper bounds in line 21 and 22, respectively, requires $\mathcal{O}(k\ell)$ operations. Adding up the cost of all individual lines and noting that $\mathcal{O}(\ell), \mathcal{O}(k) \subseteq \mathcal{O}(k\ell)$ gives the desired result.

Lemma 1 shows that the cost of computing the error bounds for the Lanczos approximations for Hermitian positive definite A is independent of n. If n is large and k and ℓ are small in comparison, the additional cost for computing the bounds is almost negligible. In the numerical experiments reported in Section 6 we demonstrate that values of k between 5 and 20 and values of ℓ between 5 and 50 are typically sufficient to compute very accurate error bounds, also for large matrix dimension n. Of course, to further reduce the additional computational work, one can compute error bounds in Algorithm 2 not in every iteration but only from time to time, e.g., once every ten iterations.

Remark 1 We conclude this section by briefly commenting on the situation when using the restarted Lanczos method (with restart length m), see, e.g., [1,12,16,32,41] instead of the full Lanczos method. In this case, one can not use the Lanczos restart recovery from Theorem 5 to compute error bounds for all iterations of the method. If m denotes the restart length, we can only compute error bounds for the first $m-k-1$ iterations of each cycle by the approach described before. However, in each restart cycle, one aims to approximate the *error* of the iterate from the last restart cycle so that the norm of the additive correction computed in cycle j can be interpreted as an estimate for the norm of the error of the iterate from cycle $j-1$ and thus gives a first hint on the progress of the method. One can now refine this using our theory to achieve upper or lower bounds for the error. To do so, one can use the result of Theorem 4 directly. When restarting the method, we perform the Lanczos process in the next, jth restart cycle with the matrix A and the vector $\boldsymbol{v}_{m+1}^{(j-1)}$. This is exactly what is needed for computing the tridiagonal matrix used in (13) for approximating $\|f(A)\boldsymbol{b} - \boldsymbol{f}_m^{(j-1)}\|$. So, in contrast to the full Lanczos method, using Theorem 4 directly does not mean additional matrix-vector products, as those of the primary and secondary Lanczos process coincide. Therefore, at the end of cycle j one can compute lower and upper bounds from an m-point Gauss and an $(m+1)$-point Gauss–Radau rule, respectively, at virtually no additional cost at all. We will give some examples for the bounds achieved this way in the numerical experiments reported in Section 6.

5 Extension to non-Hermitian matrices

We now briefly sketch how it is possible to transfer the techniques used in the previous sections to the case of non-Hermitian matrices. Most of the theoretical results concerning, e.g., the sign of the error in the outer and inner quadrature rules, do not hold any longer in this case, so that one only gets *estimates* for the error, but no bounds, in general.

In [8,13], it is shown that the Arnoldi process [3,38] can be related to quadrature rules in a similar way to how this was done for the Lanczos process in Section 2. The idea is to investigate the following quadratic form induced by A and $\boldsymbol{v}$

$$\langle h_1, h_2\rangle_{A,\boldsymbol{v}} := \boldsymbol{v}^H h_1(A)^H h_2(A)\boldsymbol{v} \tag{16}$$

for functions h_1, h_2 defined on $\operatorname{spec}(A)$. When the functions h_1 and h_2 are both analytic in a neighborhood of $\operatorname{spec}(A)$, one can rewrite (16) by the Cauchy integral formula as a double integral along a path Γ which winds around $\operatorname{spec}(A)$ exactly once,

$$\langle h_1, h_2\rangle_{A,\boldsymbol{v}} = \frac{1}{4\pi^2}\int_\Gamma\int_\Gamma \overline{h_1(z_1)} h_2(z_2)\boldsymbol{v}^H(\overline{z_1}I - A^H)^{-1}(z_2 I - A)^{-1}\boldsymbol{v}\,\overline{\mathrm{d}z_1}\,\mathrm{d}z_2. \tag{17}$$

Using the quantities from the Arnoldi decomposition for A and $\boldsymbol{v}$

$$AV_k = V_k H_k + h_{k+1,k}\boldsymbol{v}_{k+1}\boldsymbol{e}_k^T\,, \tag{18}$$

where H_k is unreduced upper Hessenberg, one approximates (16) as

$$\langle h_1, h_2\rangle_{A,\boldsymbol{v}} \approx \|\boldsymbol{v}\|^2 \boldsymbol{e}_1^H h_1(H_k)^H h_2(H_k)\boldsymbol{e}_1. \tag{19}$$

This can be interpreted as a k-point quadrature rule for (17). One can also show that the polynomials p_k defining the Arnoldi basis vectors are orthogonal with respect to the quadratic form (16), i.e.,

$$\langle p_i, p_j\rangle_{A,\boldsymbol{v}} = \delta_{i,j},$$

and consequently one has that the resulting k-point quadrature rules are exact for $(h_1, h_2) \in \mathcal{W}_{\ell-1}$, where

$$\mathcal{W}_{k-1} = (\mathit{\Pi}_{k-1} \oplus \mathit{\Pi}_k) \cup (\mathit{\Pi}_k \oplus \mathit{\Pi}_{k-1}),$$

$\mathit{\Pi}_k$ denoting the space of all polynomials of degree $\leq k$; see [8] for details. We do not present the theoretical analysis of the resulting Gauss rules, as most of this theory is not important or not applicable for the developments presented in this section. For example, the conditions given in [8] under which these *Arnoldi quadrature rules* give upper or lower bounds for the bilinear form (16) are not fulfilled in our setting.

One can now use the upper Hessenberg matrix H_k resulting from k steps of the Lanczos process applied to A and $\boldsymbol{v}_{m+1}$ to compute error estimates for the mth Arnoldi approximation to $f(A)\boldsymbol{b}$, where f is a Stieltjes function, just as in the Hermitian case, by setting $h_1 = h_2 = \widetilde{e}_m$ in (19). The key to computing error estimates with affordable computational cost in the Hermitian case was given by Theorem 5, which allows to perform the Lanczos process on a matrix of dimension at most $(2k+1) \times (2k+1)$ instead of $n \times n$. Unfortunately, the result given in [18] holds only in the Hermitian case. In the next theorem, we give a modification of Theorem 5 for non-Hermitian A, which at least allows to perform the Arnoldi process on a matrix of dimension $(m+k+1) \times (m+k+1)$ instead of A. The proof of this result is almost the same as for the original result from [18].

Theorem 6 *Let H_{m+1+k} be the upper Hessenberg matrix resulting from $k+m+1$ steps of the Arnoldi process for A and $\boldsymbol{v}_1$. Further, let $\hat{H}$ denote the upper Hessenberg matrix resulting from k steps of the Arnoldi process applied to H_{m+1+k} and $\boldsymbol{e}_{m+1}$. Then $\hat{H} = H_k^{(2)}$, where $H_k^{(2)}$ denotes the matrix resulting from k iterations of the Arnoldi process for A and $\boldsymbol{v}_{m+1}$.*

Proof Let the Arnoldi decomposition arising from k steps of Arnoldi's method for A and $\boldsymbol{v}_{m+1}$ be given as

$$A\widetilde{V}_k = \widetilde{V}_k H_k^{(2)} + h_{k+1,k}^{(2)}\widetilde{\boldsymbol{v}}_{k+1}\boldsymbol{e}_k^H. \tag{20}$$

As $\boldsymbol{v}_{m+1} \in \mathcal{K}_{m+1}(A, \boldsymbol{v}_1)$, we have that

$$\mathcal{K}_{k+1}(A, \boldsymbol{v}_{m+1}) \subseteq \mathcal{K}_{m+k+1}(A, \boldsymbol{v}_1).$$

Therefore, the basis vectors $\widetilde{\boldsymbol{v}}_1, \ldots, \widetilde{\boldsymbol{v}}_{k+1}$ generated by the Arnoldi process for A and $\boldsymbol{v}_{m+1}$ all lie in $\mathcal{K}_{m+k+1}(A, \boldsymbol{v}_1)$ and can thus be written as linear combinations of the basis vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{k+m+1}$, i.e.,

$$[\widetilde{V}_k, \widetilde{\boldsymbol{v}}_{k+1}] = V_{m+k+1}[Q_k, \boldsymbol{q}_{k+1}] \tag{21}$$

for some matrix $Q_k \in \mathbb{C}^{(m+k+1)\times k}$. As $[\widetilde{V}_k, \widetilde{\boldsymbol{v}}_{k+1}]$ and V_{m+k+1} both have orthonormal columns, $[Q_k, \boldsymbol{q}_{k+1}]$ must have orthonormal columns as well. Inserting (21) into the Arnoldi decomposition (20) gives

$$AV_{k+m+1}Q_k = V_{m+k+1}Q_k H_k^{(2)} + h_{k+1,k}^{(2)} V_{m+k+1}\boldsymbol{q}_{k+1}\boldsymbol{e}_k^H. \tag{22}$$

Left-multiplying both sides of (22) by the projector $V_{m+k+1}V_{m+k+1}^H$ onto the space $\mathcal{K}_{m+k+1}(A, \boldsymbol{v}_1)$ gives

$$V_{m+k+1}V_{m+k+1}^H AV_{k+m+1}Q_k = V_{m+k+1}Q_k H_k^{(2)} + h_{k+1,k}^{(2)} V_{m+k+1}\boldsymbol{q}_{k+1}\boldsymbol{e}_k^H.$$

which, using $V_{m+k+1}^H AV_{k+m+1} = H_{m+k+1}$ (which follows from (18)), simplifies to

$$V_{m+k+1}H_{m+k+1}Q_k = V_{m+k+1}Q_k H_k^{(2)} + h_{k+1,k}^{(2)} V_{m+k+1}\boldsymbol{q}_{k+1}\boldsymbol{e}_k^H.$$

Noting that V_{m+k+1} has full (column) rank, this implies

$$H_{m+k+1}Q_k = Q_k H_k^{(2)} + h_{k+1,k}^{(2)}\boldsymbol{q}_{k+1}\boldsymbol{e}_k^H. \tag{23}$$

By [40, Chapter 5, Theorem 1.3] the Arnoldi decomposition is unique up to scaling of the basis vectors by complex scalars of modulus one (and corresponding scaling of the entries in H_k). As all subdiagonal entries of $H_k^{(2)}$ are positive because they were computed by Arnoldi's method, it follows that (23) is the unique Arnoldi decomposition corresponding to H_{m+k+1} and $\boldsymbol{q}_1$. As $\widetilde{\boldsymbol{v}}_1 = \boldsymbol{v}_{m+1}$, we have that $\boldsymbol{q}_1 = \boldsymbol{e}_{m+1}$, which proves the result.

According to Theorem 6, for a non-Hermitian matrix A, it is not possible to only use a small sub-block of H_{m+k+1} in a secondary Arnoldi process for retrieving $H_k^{(2)}$. Therefore, the cost for computing error estimates in this way scales with the iteration number m, as multiplications with a matrix of size $(m + k + 1) \times (m + k + 1)$ have to be performed. Other than that, one can straight-forwardly adapt the approach of Algorithm 2 to the non-Hermitian case with minor modifications (replacing the Lanczos process by the Arnoldi process and the computation of the bounds in line 27 and 28 by a quadrature-based approximation for (19) and replacing the recurrence relations for the ρ_i, as the simple update formulas do not hold anymore when H_m is not tridiagonal). The resulting method is given in Algorithm 3.

In contrast to Algorithm 2, there is no guarantee that the exact error of the iterate $\boldsymbol{f}_m$ lies below the prescribed tolerance `tol` when the value `estimate` does in Algorithm 3, so that it is not always advisable to use this error estimate as a stopping criterion, especially if it is crucial that the prescribed tolerance is reached. The numerical experiments in Section 6, however, suggest that the error estimates are rather accurate in many situations, e.g., when dealing with Stieltjes functions. In such situations it may therefore often be sufficient to use a certain safety factor $\varepsilon < 1$ in the computations and run Algorithm 3 with the tolerance $\varepsilon \cdot$ `tol` instead of `tol`.

The additional cost of Algorithm 3 in comparison to the standard Arnoldi method without computation of error bounds is given in the next lemma.

Algorithm 3: Arnoldi approximation with error estimate for $f(A)\boldsymbol{b}$

```
    Given: $A$, $\boldsymbol{b}$, $f$, $k$, $\ell$, tol
1   Choose quadrature nodes/weights $(t_i, \omega_i)_{i=1,\ldots,\ell}$.    {inner quadrature rule}
2   $\boldsymbol{v}_1 \leftarrow \frac{1}{\|\boldsymbol{b}\|}\boldsymbol{b}$
3   for $m = 1, 2, \ldots$ do
4       Compute $\boldsymbol{w}_m \leftarrow A\boldsymbol{v}_m$
5       for $i = 1, \ldots, m$ do
6           $h_{i,m} \leftarrow (\boldsymbol{w}_m, \boldsymbol{v}_i)$
7           $\boldsymbol{w}_m \leftarrow \boldsymbol{w}_m - h_{i,m}\boldsymbol{v}_i$
8       $h_{m+1,m} \leftarrow \|\boldsymbol{w}_m\|$
9       if $h_{m+1,m} = 0$ then
10          compute $\boldsymbol{f}_m \leftarrow \|\boldsymbol{b}\| V_m f(H_m)\boldsymbol{e}_1$.
11          Stop.
12      $\boldsymbol{v}_{m+1} \leftarrow \frac{1}{h_{m+1,m}}\boldsymbol{w}_m$
13      for $i = 1, \ldots, \ell$ do
14          $\rho_i(m) \leftarrow \boldsymbol{e}_m^H (H_m + tI)^{-1}\boldsymbol{e}_1$
15      if $m \geq k + 1$ then
16          Perform $k$ steps of Arnoldi's method for $H$ and $\boldsymbol{e}_{m-k+1}$, yielding $\hat{H}$.
17          estimate $\leftarrow \|\boldsymbol{b}\|^2 \boldsymbol{e}_1^H \sum_{i=1}^{\ell} \omega_i |\rho_i(m-k)|^2 (\hat{H} + t_i I)^{-H} (\hat{H} + t_i I)^{-1}\boldsymbol{e}_1$
18          if estimate $\leq$ tol then
19              compute $\boldsymbol{f}_m \leftarrow \|\boldsymbol{b}\| V_m f(H_m)\boldsymbol{e}_1$.
20              Stop.
```

Lemma 2 *Performing Algorithm 3 instead of Arnoldi's method for $A \in \mathbb{C}^{n\times n}$ and $\boldsymbol{b} \in \mathbb{C}^n$ requires an additional computational cost of the order $\mathcal{O}(m^2(k+\ell) + k\ell)$ in the mth iteration and thus $\mathcal{O}(m_{\max}^3(k+\ell) + m_{\max}k\ell)$, if $m_{\max}$ iterations are necessary to reach the desired accuracy. In particular, the cost per iteration is independent of n, but not of m.*

Proof The proof is very similar to the one of Lemma 1, with the following differences. The secondary Arnoldi process for step m now has a cost of $\mathcal{O}(m^2k)$, since each multiplication with H_{m+k+1} has cost $\mathcal{O}(m^2)$ as the upper triangle of this matrix is in general dense (and we assume $k \in \mathcal{O}(m)$). The solution of each linear system in line 14 has cost $\mathcal{O}(m^2)$ due to the Hessenberg structure of H_m, which results in $\mathcal{O}(m^2\ell)$ for all systems.

We note that while according to Lemma 2, the cost of computing error bounds in Algorithm 3 grows with the number of iterations that are performed, the cost of the algorithm is still dominated by the matrix vector product and the orthogonalization process in the $\mathcal{O}$-sense: The cost of the orthogonalization against the previous basis vectors in the mth iteration of Algorithm 3 is of order $\mathcal{O}(mn)$, and we have that $\mathcal{O}(m^2k) \subseteq \mathcal{O}(mn)$ if k is fixed independently of m and n. Again, it might nonetheless be attractive to only compute error estimates after some, but not all of the iterations of the method.

6 Numerical experiments

In this section, we compute error bounds and estimates for Lanczos/Arnoldi approximations for a few model problems. We begin with investigating examples where the

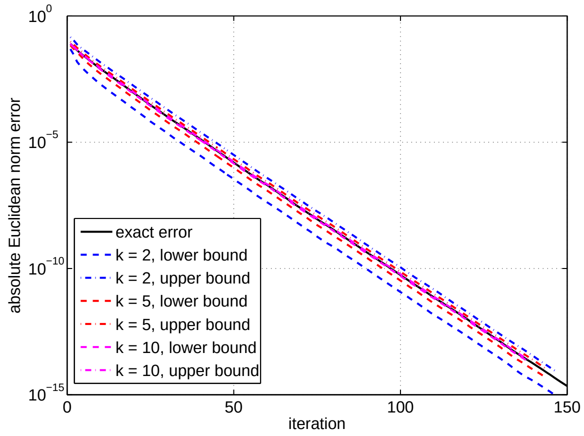

Fig. 1 Exact error norm and bounds computed by Algorithm 2 for approximating $A^{-1/2}\boldsymbol{z}$ in the Gaussian Markov random field model problem. The inner quadrature rule uses $\ell = 20$ nodes, while the number of nodes in the outer quadrature rule is varied between $k = 2, 5$ and 10.

theory from Section 3 applies, i.e., when approximating Stieltjes matrix functions of Hermitian positive definite matrices.

The first model problem we consider is sampling from a Gaussian Markov random field, see, e.g., [32, 39]. Given a set of n points $s_i \in \mathbb{R}^d, = 1, \ldots, n$, one defines the *precision matrix* $A \in \mathbb{R}^{n \times n}$ (with respect to two parameters δ, ϕ) of these points as

$$a_{ij} = \begin{cases} 1 + \phi \sum_{k=1,k\neq i}^{n} \chi_{ij}^{\delta} & \text{if } i = j, \\ -\phi \chi_{ij}^{\delta} & \text{otherwise,} \end{cases}$$

where χ^{δ} is given by

$$\chi_{ij}^{\delta} = \begin{cases} 1 & \text{if } \|s_i - s_j\| < \delta, \\ 0 & \text{otherwise.} \end{cases}$$

The precision matrix A is Hermitian and strictly diagonally dominant, its smallest eigenvalue being 1 (corresponding to the eigenvector $(1, \ldots, 1)^H$). A sample from a Gaussian Markov random field (which is a collection of random variables x_i corresponding to the points s_i) is obtained by computing $A^{-1/2}\boldsymbol{z}$, where $\boldsymbol{z}$ is a vector of independently and identically distributed standard normal random variables. We simulate $n = 50{,}000$ points in the unit square with parameters $\phi = 3, \delta = 0.01$, resulting in $\operatorname{spec}(A) \subset [1, 109.6]$ (i.e., a rather well-conditioned matrix) and 830,626 nonzeros in A. As $A^{-1/2}$ is a Stieltjes matrix function and A is positive definite, Theorem 4 applies and we can compute lower and upper bounds for the error norm in Algorithm 2.

Figure 1 presents the bounds computed by Algorithm 2 for the Gaussian Markov random field model problem for 100 Lanczos iterations and different values of k. We show the exact error norm as well as the lower and upper bounds computed with $k = 2, 5$ and 10 quadrature nodes for the outer Gauss and Gauss–Radau rule. Recall that the smallest eigenvalue of A is explicitly known to be 1, so that we fixed the leftmost node of the Gauss–Radau rule to this value. For the inner quadrature rule

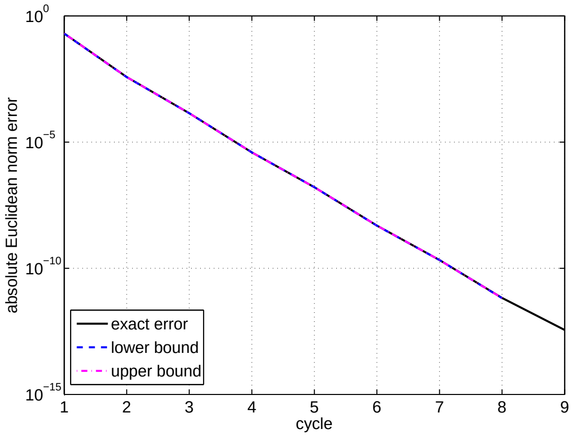

Fig. 2 Exact error norm and bounds computed in the restarted Lanczos method for approximating $A^{-1/2}\boldsymbol{z}$ in the Gaussian Markov random field model problem. The inner quadrature rule uses $\ell = 20$ nodes, the number of outer quadrature nodes corresponds to the restart length $m = 20$.

used for evaluating the error function we use $\ell = 20$ nodes of a Gauss and Gauss–Radau rule, respectively, chosen such that the sign of the error in the inner and outer quadrature rule is the same, and we compute guaranteed bounds, cf. also Proposition 2. As in this model problem, the quality of the bounds is very insensitive to the value of ℓ and does not substantially change when ℓ is increased, we do not report results for other values. A study of the influence of ℓ is given in the next experiment.

For all values of k, we observe that the qualitative behavior of the error is captured accurately and that for this well-conditioned matrix even for the very small number of $k = 2$ quadrature nodes, the error is only overestimated (respectively underestimated) by about one order of magnitude. For $k = 5$ and $k = 10$, the error bounds lie very close to the exact value of the integral, however they are only available three (respectively eight) iterations later than the bounds for $k = 2$. This raises the question of which value for k is "optimal" in the sense that it allows to terminate the iteration as early as possible. For small values of k the bounds are more inaccurate but available early, and for large values of k the bounds are very accurate but available later. Therefore, it is not clear *a priori* which value of k in this trade-off between accuracy and early availability is optimal, as this does not only depend on the the quality of the bounds with respect to k, but also, among other factors, on the steepness of the convergence slope for the given function, matrix and right-hand side. When trying to achieve a tolerance $\texttt{tol} = 10^{-9}$ in this experiment, the Lanczos algorithm can be stopped after 92 iterations for $k = 2$ and $k = 5$ and after 96 iterations for $k = 10$, despite the bounds for $k = 10$ being extremely accurate. We investigate this topic more thoroughly in the next numerical experiment.

In Figure 2, we report bounds for the *restarted* Lanczos method computed according to Remark 1. We use restart length $m = 20$, so that the computed bounds correspond to 20-point Gauss and 21-point Gauss–Radau rules, respectively. Both bounds are almost indistinguishable from the exact error norm, which might be expected due

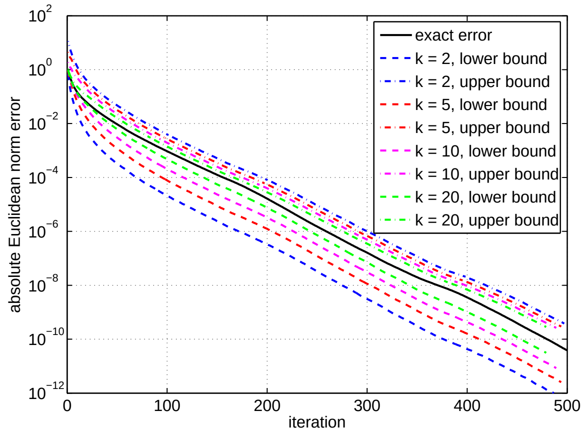

Fig. 3 Exact error norm and bounds computed by Algorithm 2 for approximating $(Q^2)^{-1/2}Q\boldsymbol{b}$ in the Hermitian QCD model problem. The inner quadrature rule uses at most $\ell = 100$ nodes, while the number of nodes in the outer quadrature rule is varied between $k = 2, 5, 10$ and 20.

to the rather high number of quadrature nodes used. This example illustrates that it is very attractive to use the developed error bounds also in the restarted Lanczos method, especially considering that the additional work which is necessary for computing them is even less than for the unrestarted Lanczos method, as no secondary Lanczos process is necessary.

The second model problem we consider is from the area of *quantum chromodynamics* (QCD), a branch of Theoretical Physics in which the strong interaction between quarks is studied. In *Lattice QCD*, this theory is discretized and simulated on a four-dimensional space–time lattice (with 12 variables per lattice point, corresponding to combinations of three colors and four spins). For computing certain observables, it is necessary that the *chiral symmetry* is preserved on the lattice. This can be achieved by simulating *overlap fermions*, which requires the solution of linear systems involving the overlap Dirac operator (see [35])

$$N_{\mathrm{ovl}} := \rho I + \mathit{\Gamma}_5 \,\mathrm{sign}(Q), \tag{24}$$

where $\rho > 1$ is a mass parameter, the matrix Q is the symmetrized Wilson kernel, see [17], e.g., representing a periodic nearest-neighbor coupling on the lattice, and $\mathit{\Gamma}_5$ is a permutation which permutes the spins on each lattice point in an identical manner. The matrix Q is sparse and complex. Depending on a *chemical potential* ν, the matrix Q is Hermitian ($\nu = 0$) or non-Hermitian ($\nu > 0$). As one can not explicitly compute the matrix sign(Q) for realistic grid sizes, one typically solves (24) by an iterative Krylov subspace method which only requires performing matrix-vector products with sign(Q). At each outer Krylov iteration one therefore has to compute sign$(Q)\boldsymbol{b}$ where $\boldsymbol{b}$ changes from one iteration to the next. One typically computes the sign function via

$$\mathrm{sign}(Q)\boldsymbol{b} = (Q^2)^{-1/2}Q\boldsymbol{b};$$

see, e.g., [42]. Therefore, at zero chemical potential, approximating $\mathrm{sign}(Q)\boldsymbol{b}$ amounts to applying the inverse square root of the Hermitian positive definite matrix Q^2 to the vector $Q\boldsymbol{b}$, so that our theory for computing error bounds applies again. In our experiment, we use a lattice with 8^4 points, resulting in a matrix Q of dimension $12 \cdot 8^4 = 49{,}152$ with condition number about $3 \cdot 10^3$. We again report the exact error norm and the bounds computed for different values of k in Figure 3. For computing the upper bounds for the error, it is necessary to known the smallest eigenvalue of Q (or at least a good approximation of it). In this experiment, we do not know the smallest eigenvalue explicitly. As it is quite costly to approximate the eigenvalues of Q before starting the Lanczos iteration, we use the smallest Ritz value as an approximation to $\lambda_{\min}$, as soon as it does not change substantially any longer from one iteration to the next. We then assume that it has converged to $\lambda_{\min}$ to sufficient accuracy and use it (multiplied with the safety factor 0.99) as the fixed quadrature node at the left of the interval of integration for the Gauss–Radau rule. This approach was suggested in [18]. Qualitatively, the results obtained in this experiment are similar to those for the Gaussian Markov random field model problem, in the sense that the bounds again capture the behavior of the exact error norm very well. The order of magnitude, however, is not captured as well as before, i.e., the bounds are not as close to the exact error norm. Especially the lower bounds underestimate the exact error norm by a quite large margin for the smaller values of k (e.g., about two orders of magnitude for $k = 2$). The upper bounds lie closer to the actual error norm also for small values of k and do not improve by such a large margin when k is increased.

Another difference between this and the previous experiment which is worth mentioning is that larger numbers ℓ of inner quadrature nodes are necessary to achieve satisfactory bounds. The results reported in Figure 3 were produced using an adaptive approach where two quadrature rules of different order $\ell, \widetilde{\ell}$ are used to estimate the quadrature error (and increase the number of nodes if the error estimate is to large). To illustrate the effect that the value ℓ can have when it is chosen too small, a comparison of different values of ℓ is given in Table 1, where we report the maximum of the ratio between the bounds computed for $\ell = 10, 20, 50$ and the bound for $\ell = 100$ (in all cases, $k = 5$). The ratios between the quadrature rules for $\ell = 10$ and $\ell = 20$ and the most accurate tested rule for $\ell = 100$ are very large, showing that the approximations computed for these values are not reasonable and one can expect the error in the inner quadrature rule to make a non-negligible contribution to (or even dominate) the quality of the computed bounds. The ratio between $\ell = 50$ and $\ell = 100$ is still not small, but in most cases acceptable, as the deviation of the bound from the exact error will often not be dominated by it (as the outer Gauss rules are typically more inaccurate).

We also give a more thorough study on what the optimal value of k is for stopping the iteration as early as possible. Figure 4 reports the iteration numbers at which Algorithm 2 is terminated for different values of k when a tolerance of $\texttt{tol} = 10^{-9}$ is required. The optimal values in this case are found to be $k = 12$ or 13. For smaller or larger values, a higher number of iterations is necessary. For values of k larger than 20, the increase in the number of necessary iterations is almost proportional to the increase in k. This behavior is quite intuitive. For large values of k, the bounds are already quite accurate, and further increasing the number of quadrature nodes does not improve them by much. However, each additional quadrature node delays the iteration at which convergence is detected. Therefore, it seems like a reasonable guideline for practical computations to choose not too many quadrature nodes, say, not more than $k = 20$, for the outer quadrature rule, in particular when considering that the number

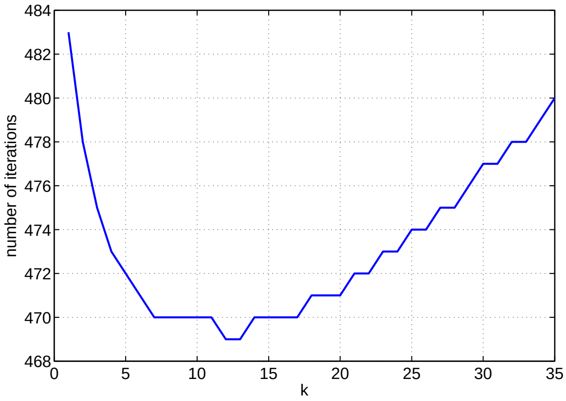

Fig. 4 Iteration number at which the stopping criterion in Algorithm 2 is fulfilled (with $\texttt{tol} = 10^{-9}$) for the Hermitian QCD model problem and different values of k.

ℓ	ratio lower bound	ratio upper bound
10	5500	6.18
20	10.47	3.17
50	1.12	1.33

Table 1 Maximum ratio between the of the bounds computed by Algorithm 2 for the values $\ell = 10, 20, 50$ of inner quadratures nodes and the bounds computed for $\ell = 100$ for the Hermitian lattice QCD model problem and $k = 5$.

of superfluous iterations is not very large for the smaller values of k (e.g., for $k = 5$, one performs 472 instead of 469 iterations).

We again also present error bounds computed in the *restarted* Lanczos method (with restart length $m = 20$) for the Neuberger overlap operator at zero chemical potential, see Figure 5. The upper bound again almost completely agrees with the exact error norm, the lower bound slightly underestimates it, but by less than one order of magnitude, again demonstrating that this approach gives very accurate estimates for the error norm in the restarted Lanczos setting.

As a variation of the previous experiment, we consider approximating the action of the overlap operator at *nonzero chemical potential*, which leads to computing the inverse square root of a non-Hermitian matrix. We use a matrix of the same size as in the zero chemical potential case, but introduce a chemical potential $\nu = 1/20$. We refer to [6], e.g., for a description how ν enters into the Wilson kernel Q. The error estimates computed by Algorithm 3 are depicted in Figure 6. The error estimates again show the same qualitative behavior as the exact error norm. Nonetheless, the error is severely underestimated for smaller values of k. The estimate for $k = 2$ differs from the exact value by about two orders of magnitude, for $k = 10$, the error is underestimated by about one order of magnitude. Although we do not have theoretical results on the behavior or quality of the error estimates in the non-Hermitian case, this experiment at least illustrates that there are situations in which one can expect the estimates to reasonably capture the convergence behavior of the method. We do not report results for the restarted Arnoldi method, since in the non-Hermitian case, where one can not

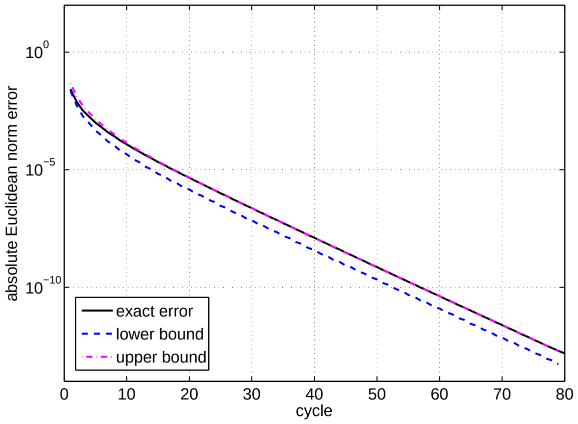

Fig. 5 Exact error norm and bounds computed by the restarted Lanczos method for approximating $(Q^2)^{-1/2}Q\boldsymbol{b}$ in the Hermitian lattice QCD model problem. The inner quadrature rule uses at most $\ell = 50$ nodes, the number of outer quadrature nodes corresponds to the restart length $m = 20$.

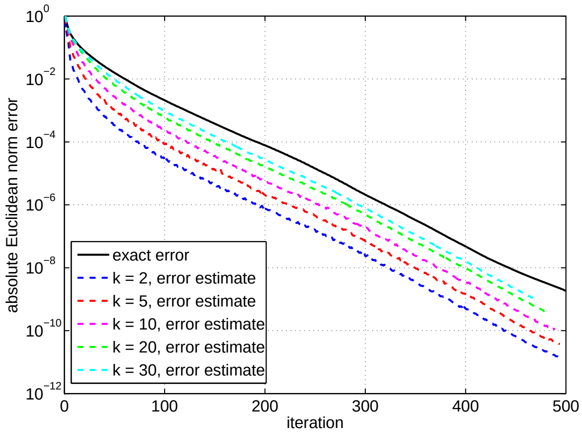

Fig. 6 Exact error norm and estimates computed by Algorithm 3 for approximating $(Q^2)^{-1/2}Q\boldsymbol{b}$ in the non-Hermitian QCD model problem. The inner quadrature rule uses $\ell = 100$ nodes, while the number of nodes in the outer quadrature rule is varied between $k = 2, 5, 10, 20$ and 30.

expect to compute bounds, there is no reason to expect the quadrature based estimates to be better approximations to the exact error norm than just taking the norm of the last update as an estimate.

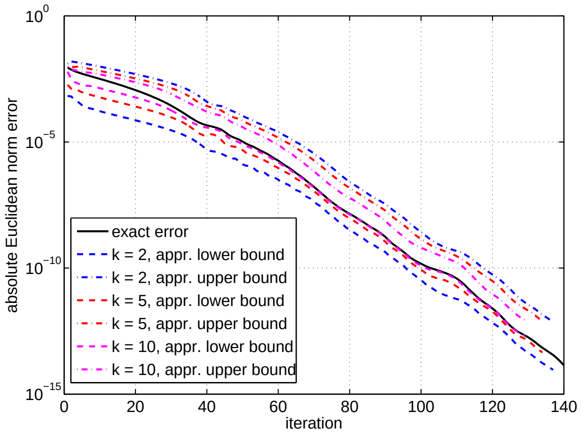

Fig. 7 Exact error norm and (approximate) bounds computed by Algorithm 2 for approximating $e^{-\theta\sqrt{A}}A^{-1}\boldsymbol{b}$ for the semi-discretization of the wave equation. The inner quadrature rule uses $\ell = 20$ nodes, while the number of nodes in the outer quadrature rule is varied between $k = 2, 5$ and 10.

In our last experiment, taken from [11], we consider the semi-discretization of the wave equation

$$\begin{aligned} \Delta u - \frac{\partial^2 u}{\partial \theta^2} &= 0 \qquad \text{on } (0,1)^3 \times (0,\infty), \\ u(x,t) &= 0 \qquad \text{on } \partial(0,1)^3 \text{ for all } \theta \in [0,\infty), \\ u(x,0) &= u_0(x) \text{ for all } x \in (0,1)^3. \end{aligned}$$

The discretized equation is solved by

$$\boldsymbol{u}(\theta) = e^{-\theta\sqrt{A}}\boldsymbol{u}_0,$$

where A is the standard finite difference discretization of the Laplace operator. We rewrite this as

$$\boldsymbol{u}(\theta) = (I - Af(A))\boldsymbol{u}_0,$$

with

$$f(z) = \frac{1 - e^{-\theta\sqrt{z}}}{z}, \tag{25}$$

and note that the function f from (25) has the integral representation

$$f(z) = \int_0^\infty \frac{1}{t+z}\frac{\sin(\theta\sqrt{t})}{\pi t}\,\mathrm{d}t.$$

While f from (25) is *not* a Stieltjes function (as the corresponding function μ satisfies $\mu'(t) = \sin(\theta\sqrt{t})/(\pi t)$ and thus is not monotonically increasing), we can still use the algorithmic techniques from Section 4 for computing error estimates, which are, however, no guaranteed bounds any longer. In this experiment, we choose $N = 51$ grid points in each spatial direction, $\theta = 0.1$ and initial conditions $\boldsymbol{u}_0 = \mathbf{1}$. Figure 7 provides results for both Gauss and Gauss–Radau quadrature, which show that we still get bounds in this case, although we can not rely on this to be true. For the Gauss–Radau rule, we use the fact that the smallest eigenvalue of the three-dimensional discretized Laplacian is explicitly known and thus cheaply available. Alternatively, one could again use the smallest Ritz value after some iterations as an approximation, giving very similar results. The inner quadrature in this case is a standard 20-point Gauss–Legendre rule. The quality of the computed estimates is very similar to what was already observed in the two other experiments involving Hermitian matrices. The (approximate) upper error bound decreases below 10^{-9} in iteration 108 for $k = 2$ and $k = 5$ and in iteration 109 for $k = 10$, so that again all values lie closely together and each of them represents a reasonable choice. However, we again stress that one has to keep in mind that in this situation, the error estimate decreasing below 10^{-9} is not a guarantee that the exact error norm lies below the tolerance (although this is the case in the example presented here), so that one has to be careful when using it as a sole stopping criterion in an actual computation.

7 Conclusions

We presented an approach to compute quadrature based estimates for the norm of the error of (restarted) Arnoldi/Lanczos approximations for $f(A)\boldsymbol{b}$ when f is a Stieltjes function. When A is Hermitian positive definite, these error estimates can be computed essentially for free (with cost independent of the matrix size n and the iteration number m) by using the concept of Lanczos restart recovery from [18] together with recurrence relations for the residual norms of Lanczos iterates for shifted linear systems. We proved that for suitably chosen quadrature rules the computed estimates are upper or lower bounds for the exact error norm when A is Hermitian positive definite. Numerical experiments with matrices from practical applications illustrated the quality of the bounds as well as their dependence on certain parameters. For non-Hermitian A it can not be guaranteed that the estimates are bounds and Lanczos restart recovery is not applicable. We did, however, propose a variant of this restart recovery which allows to also compute error estimates in the non-Hermitian case with additional cost which, although not being independent of the iteration number m, is lower than the orthogonalization cost in the Arnoldi process.

References

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. H. Alzer and C. Berg, *Some classes of completely monotonic functions*, Ann. Acad. Sci. Fenn. Math., 27 (2002), pp. 445–460.
3. W. E. Arnoldi, *The principle of minimized iteration in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.
4. C. Berg, *Stieltjes-Pick-Bernstein-Schoenberg and their connection to complete monotonicity*, in Positive Definite Functions. From Schoenberg to Space-Time Challenges, S. Mateu and E. Porcu, eds., Dept. of Mathematics, University Jaume I, Castellón de la Plana, Spain, 2008.
5. C. Berg and G. Forst, *Potential Theory on Locally Compact Abelian Groups*, Springer-Verlag, Heidelberg, 1975.

6. J. Bloch, A. Frommer, B. Lang, and T. Wettig, *An iterative method to compute the sign function of a non-Hermitian matrix and its application to the overlap Dirac operator at nonzero chemical potential*, Comput. Phys. Commun., 177 (2007), pp. 933–943.
7. K. Burrage, N. Hale, and D. Kay, *An efficient implicit FEM scheme for fractional-in-space reaction–diffusion equations*, SIAM J. Sci. Comput., 34 (2012), pp. A2145–A2172.
8. D. Calvetti, S. Kim, and L. Reichel, *Quadrature rules based on the Arnoldi process*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 765–781.
9. P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration*, Academic Press, New York, 1975.
10. V. Druskin and L. Knizhnerman, *Two polynomial methods of calculating functions of symmetric matrices*, Comput. Math. Math. Phys., 29 (1989), pp. 112–121.
11. 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. 775–771.
12. 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.
13. R. W. Freund and M. Hochbruck, *Gauss quadrature associated with the Arnoldi process and the Lanczos algorithm*, in Linear algebra for large scale and real-time applications, M. S. Moonen, G. H. Golub, and B. De Moor, eds., Kluwer, Dordrecht, 1993, pp. 377–380.
14. A. Frommer, *Monotone convergence of the Lanczos approximations to matrix functions of Hermitian matrices*, Electron. Trans. Numer. Anal., 35 (2009), pp. 118–128.
15. 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.
16. 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.
17. A. Frommer, K. Kahl, S. Krieg, B. Leder, and M. Rottman, *Adaptive Aggregation Based Domain Decomposition Multigrid for the Lattice Wilson Dirac Operator*, SIAM J. Sci. Comput., 36 (2014), pp. A1581–A11608.
18. A. Frommer, K. Kahl, T. Lippert, and H. Rittich, *2-norm error bounds and estimates for Lanczos approximations to linear systems and rational matrix functions*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 1046–1065.
19. A. Frommer and P. Maass, *Fast CG-based methods for Tikhonov–Phillips regularization*, SIAM J. Sci. Comput., 20 (1999), pp. 1831–1850.
20. A. Frommer and V. Simoncini, *Stopping criteria for rational matrix functions of hermitian and symmetric matrices*, SIAM J. Sci. Comput., 30 (2008), pp. 1387–1412.
21. ———, *Error bounds for Lanczos approximations of rational functions of matrices*, in Numerical Validation in Current Hardware Architectures, A. Cuyt, W. Krmer, W. Luther, and P. Markstein, eds., vol. 5492 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2009, pp. 203–216.
22. E. Gallopoulos and Y. Saad, *Efficient solution of parabolic equations by Krylov approximation methods*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 1236–1264.
23. W. Gautschi, *Orthogonal Polynomials: Computation and Approximation*, Oxford University Press, Oxford, 2004.
24. G. H. Golub and G. Meurant, *Matrices, moments and quadrature*, in Numerical analysis 1993, D. Griths and G. W. Eds., eds., vol. 303 of Pitman Research Notes in Mathematics Series, Longman Scientific & Technical, Harlow, (1994), pp. 105–156.
25. ———, *Matrices, moments and quadrature. II. How to compute the norm of the error in iterative methods*, BIT, (1997), pp. 687–705.
26. ———, *Matrices, Moments and Quadrature with Applications*, Princeton Series in Applied Mathematics, Princeton University Press, Princeton and Oxford, 2010.
27. P. Henrici, *Applied and Computational Complex Analysis, Vol. 2*, John Wiley & Sons, New York, 1977.
28. M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, Journal of Research of the National Bureau of Standards, 49 (1952), pp. 409–436.
29. N. J. Higham, *Functions of Matrices: Theory and Computation*, SIAM, 2008.
30. M. Hochbruck and C. Lubich, *On Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., 34 (1997), pp. 1911–1925.
31. M. Hochbruck and A. Ostermann, *Exponential integrators*, Acta Numer., 19 (2010), pp. 209–286.

32. 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.
33. L. Knizhnerman, *Calculation of functions of unsymmetric matrices using Arnoldi's method*, Comput. Math. Math. Phys., 31 (1991), pp. 1–9.
34. C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators.*, Journal of research of the National Bureau of Standards, 45 (1950), pp. 255–282.
35. H. Neuberger, *Exactly massless quarks on the lattice*, Phys. Lett. B, 417 (1998), pp. 141–144.
36. P. Novati, *Numerical approximation to the fractional derivative operator*, Numer. Math., 127 (2014), pp. 539–566.
37. Y. Saad, *Analysis of some Krylov subspace approximations to the exponential operator*, SIAM J. Numer. Anal., 29 (1992), pp. 209–228.
38. ———, *Iterative Methods for Sparse Linear Systems, 2nd ed.*, SIAM, 2003.
39. D. P. Simpson, *Krylov subspace methods for approximating functions of symmetric positive definite matrices with applications to applied statistics and anomalous diffusion*, PhD thesis, Queensland University of Technology, 2008.
40. G. W. Stewart, *Matrix Algorithms Volume II: Eigensystems*, SIAM, 2001.
41. 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.
42. J. van den Eshof, A. Frommer, T. Lippert, K. Schilling, and H. A. van der Vorst, *Numerical methods for the QCD overlap operator, I. Sign-function and error bounds*, Comput. Phys. Commun., 146 (2002), pp. 203–224.
43. J. van den Eshof and M. Hochbruck, *Preconditioning Lanczos approximations to the matrix exponential*, SIAM J. Sci. Comput., 27 (2006), pp. 1438–1457.