Error bounds for Lanczos approximations of rational functions of matrices

Andreas Frommer[1] and Valeria Simoncini[2]

[1] Fachbereich Mathematik und Naturwissenschaften, Bergische Universität
Wuppertal, D-42097 Wuppertal, Germany
frommer@math.uni-wuppertal.de,

[2] Dipartimento di Matematica, Università di Bologna, Piazza di Porta S. Donato, 5,
I-40127 Bologna, Italy and CIRSA, Ravenna, Italy
valeria@dm.unibo.it

Abstract. Having good estimates or even bounds for the error in computing approximations to expressions of the form $f(A)v$ is very important in practical applications. In this paper we consider the case that A is Hermitian and that f is a rational function. We assume that the Lanczos method is used to compute approximations for $f(A)v$ and we show how to obtain a posteriori upper and lower bounds on the ℓ_2-norm of the approximation error. These bounds are computed by minimizing and maximizing a rational function whose coefficients depend on the iteration step. We use global optimization based on interval arithmetic to obtain these bounds and include a number of experimental results illustrating the quality of the error estimates.

1 Introduction

Today, matrix functions are used in a large number of application problems, and the theoretical understanding of numerical methods for their computation is of topical interest. We refer to the recent book of Higham [10] as a survey reference. Usually, the function $f(A) \in \mathbb{R}^{n\times n}$ of a matrix $A \in \mathbb{R}^{n\times n}$ will be a full matrix even when A is sparse. This prevents $f(A)$ to be computed directly when n becomes large, as it is common in many applications. Fortunately, though, it is then usually sufficient to compute the action of the matrix function on a vector, i.e., $f(A)v$ for $v \in \mathbb{R}^n$, which is the task we are considering in this paper.

A prominent example where such computations arise is in exponential integrators. Here, the action of the matrix exponential $\exp(A)v$ or of $\varphi(A)v$ with $\varphi(t) = (\exp(t) - 1)/t$ must be computed. Exponential integrators have recently emerged for numerically solving stiff or oscillatory systems of ordinary differential equations; see, e.g., [11], [8]. They can also be used for the integration of the time-dependent Schrödinger equation in quantum mechanics in which case one uses trigonometric functions rather than the exponential; see [7]. Another example arises in lattice gauge theory where so-called chiral overlap fermions are simulated using a Monte-Carlo approach. In each step one has to solve linear systems of the form $(P + \text{sign}(A))x = b$, where P is a permutation matrix and A is the Wilson fermion matrix; see [4]. When solving $(P+\mathrm{sign}(A))x=b$ with an iterative method, each step will usually require the computation of $(P+\mathrm{sign}(A))p$ and thus of $\mathrm{sign}(A)p$ for a vector p which changes at each iteration.

In general, for any square matrix A, the matrix function $f(A)$ can be defined for a sufficiently smooth function f by means of the Jordan canonical form of A; see, e.g., [13]. In this paper we are only concerned with the situation where the matrix A is Hermitian. Then $f(A)$ is defined as soon as f is defined on $\mathrm{spec}(A)$, the set of all eigenvalues of A. Many equivalent definitions for $f(A)$ can be given. One is to take $f(A)=p(A)$ where p is the polynomial that interpolates f on $\mathrm{spec}(A)$. Alternatively, let $A=V\Lambda V^*$ denote the spectral decomposition of A where the columns of the orthogonal matrix V represent eigenvectors of A and the diagonal entries of the diagonal matrix Λ the corresponding eigenvalues λ_i. Then we can put

$$f(A)=Vf(\Lambda)V^* \text{ where } f(\Lambda)=\mathrm{diag}(f(\lambda_1),\ldots,f(\lambda_n)).$$

In the case of many functions such as the exponential, the sign, the square-root and trigonometric functions, a particularly attractive approach for large matrices is to use a rational function approximation

$$f(t)\approx g(t)=\frac{p_{s_1}(t)}{p_s(t)},$$

where $p_i(t)$ are polynomials of degree i; see, e.g., [3], [20]. The built-in Matlab ([15]) function for the matrix exponential, for example, uses a Padé rational approximation. Rational functions may be conveniently employed in a matrix context by using a partial fraction expansion. Assuming that there are no multiple poles, we then have

$$g(t)=\frac{p_{s_1}(t)}{p_s(t)}=p_{s_2}(t)+\sum_{i=1}^{s}\omega_i\frac{1}{t-\sigma_i}. \tag{1}$$

Since the computation of $p_{s_2}(A)b$ is trivial, we assume from now on that $p_{s_2}=0$, and concentrate on the sum representing the fractional part. When applied to a matrix A, this gives

$$z=g(A)v=\sum_{i=1}^{s}\omega_i(A-\sigma_iI)^{-1}v=\sum_{i=1}^{s}\omega_ix_i. \tag{2}$$

Since we assume the problem dimension to be large, the solutions x_i to the systems $(A-\sigma_iI)x_i=v$ must be approximated using an iterative technique. The iterative method we consider here is the Lanczos method which will be described in detail in section 2. Denoting $x_i^{(k)}$ the k-th iterate for the system $(A-\sigma_iI)x_i=v$, we get an overall approximation to $z=g(A)v$ as

$$z^{(k)}=\sum_{i=1}^{s}\omega_ix_i^{(k)}. \tag{3}$$

The aim of this paper is to present a method which obtains lower and upper bounds for

$$\|g(A)v - z^{(k)}\|, \tag{4}$$

the Euclidean norm of the error of the Lanczos approximation. Such bounds are important in computational practice because they can be used as a stopping criterion for the Lanczos process. We obtain the bounds as global minima and maxima of certain (one-dimensional) rational functions. For their computation we use a global optimization algorithm based on interval arithmetic. Note that the cost of this method does not depend on the matrix dimension n, but only on the number s of poles in the rational function and the width of the spectrum of A.

The rest of this paper is organized as follows: In section 2 we review some important facts for the Lanczos process and the Lanczos approximations to families of shifted linear systems as well as to rational matrix functions. Section 3 explains how to obtain a posteriori bounds on the error, and section 4 exposes the global optimization algorithm based on interval arithmetic that we use to compute these bounds. Finally, section 5 contains a full algorithmic description of the Lanczos method including the computation of the error bounds as well as several numerical results illustrating the quality of the error bounds.

2 The Lanczos approximation

Given a vector v such that $\|v\| = 1$ and a Hermitian matrix A, the Lanczos process generates a sequence of orthonormal vectors that span the Krylov subspace $\mathcal{K}_k(A, v) = \text{span}\{v, Av, \ldots, A^{k-1}v\}$. As k grows, the subspaces are nested, that is $\mathcal{K}_k(A, v) \subseteq \mathcal{K}_{k+1}(A, v)$. Therefore, by denoting with $\{v^{(0)}, \ldots, v^{(k-1)}\}$ the generated orthonormal basis of $\mathcal{K}_k(A, v)$, with $v^{(0)} = v$, the next vector $v^{(k)}$ such that $v^{(0)}, \ldots, v^{(k)}$ span $K_{k+1}(A, v)$ is given as

$$v^{(k)}\beta_{k+1} = Av^{(k-1)} - \alpha_k v^{(k-1)} - \beta_k v^{(k-2)}.$$

The coefficients α_k, β_k, $k = 1, 2, \ldots$ are computed so that $(v^{(j)})^* v^{(i)} = \delta_{j,i}$. Setting $V_k = [v^{(0)}, \ldots, v^{(k-1)}]$, the recurrence above can be written in compact form as

$$AV_k = V_k T_k + v^{(k)}\beta_{k+1}e_k^T, \qquad T_k = \begin{pmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_k \\ & & \beta_k & \alpha_k \end{pmatrix}, \tag{5}$$

where e_k is the kth column of the identity matrix whose dimension will be clear from the context. An approximation to the solution of the linear system $Ax = v$ may be obtained in $K_k(A, v)$ as $x_k = V_k y_k$, where y_k is obtained by imposing that the residual $r_k = v - AV_k y_k$ be orthogonal to the space, namely $V_k^* r_k = 0$. Assuming that T_k is nonsingular, explicitly writing this condition yields

$$y_k = T_k^{-1} e_1,$$

where $e_1 = (1, 0 \ldots, 0)^\mathrm{T} \in \mathbb{R}^k$.

In the context of solving shifted systems $(A - \sigma I)x = v$, we will need the following key features of the Lanczos procedure, see, e.g. [19],[24].

Lemma 1. *With the notation above, and $T_k(\sigma)$ the tridiagonal matrix from (5) with A replaced by $A - \sigma I$*

1. *For any $\sigma \in \mathbb{C}$, $K_k(A, v) = K_k(A - \sigma I, v)$ and $T_k(\sigma) = T_k - \sigma I$.*
2. *For the residual $r^{(k)}(\sigma) = v - (A - \sigma I)V_k y_k(\sigma)$ it holds*

$$r^{(k)}(\sigma) = (-1)^k \rho^{(k)}(\sigma) v^{(k)}.$$

Denoting $\theta_\nu^{(k)}, \nu = 1, \ldots, k$ the eigenvalues of T_k, we actually have

$$\rho^{(k)}(\sigma) = \prod_{\nu=1}^{k} \frac{1}{1 - \sigma/\theta_\nu^{(k)}},$$

as well as

$$\rho^{(k)}(\sigma) = (e_k^T T_k(\sigma)^{-1} e_1) \cdot \beta_k.$$

The first result shows that when solving systems that only differ for the shifting parameter σ, approximations can be carried out in a single approximation space. The second result says that the residuals associated with the shifted systems are all collinear to the next basis vector. Note that the Ritz values θ_ν are all real, since $T_k = V_k^* A V_k$ is Hermitian.

3 Error bounds

The approach we propose here to bound the error between $z = g(A)v$ and its Lanczos approximation $z^{(k)}$ requires the a priori knowledge of an interval enclosing the spectrum of A, i.e. we assume that we know ℓ_1, ℓ_2 such that $\mathrm{spec}(A) \subseteq [\ell_1, \ell_2]$. Our crucial observation starts from (3) which gives

$$g(A)v - z^{(k)} \;=\; \sum_{i=1}^{s} \omega_i \left((A - \sigma_i I)^{-1} b - x_i^{(k)} \right) \;=\; \sum_{i=1}^{s} \omega_i (A - \sigma_i I)^{-1} r_i^{(k)}.$$

Since the $x_i^{(k)}$ arise from the Lanczos process, using Lemma 1 and the notation $\rho_i^{(k)} = \rho^{(k)}(\sigma_i)$, gives

$$g(A)v - z^{(k)} = \sum_{i=1}^{s} (-1)^k \rho_i^{(k)} \omega_i (A - \sigma_i I)^{-1} v^{(k)}. \tag{6}$$

Herein, $\|v^{(k)}\| = 1$. So the error can be expressed as the action of a rational matrix function $\mathcal{R}^{(k)}(A)$, namely

$$\mathcal{R}^{(k)}(A) = \sum_{i=1}^{s} (-1)^k \rho_i^{(k)} \omega_i (A - \sigma_i I)^{-1} \tag{7}$$

on the vector $v^{(k)}$. Some additional discussion of the partial fraction expansions used is in order here. We are interested in matrix functions $f(A)$ where f is real on the real axis. It is thus natural that the rational function g which we use to approximate f is real on the real line, too. Its partial fraction expansion (1), however, might have complex poles which then come in complex conjugate pairs $\sigma, \overline{\sigma}$ and corresponding complex conjugate coefficients $\omega, \overline{\omega}$. This is the case for instance when f is the exponential function and g is a Padé or Chebyshev rational approximation. For computing our error bounds it will turn out useful to have a *real* partial fraction expansion for the rational functions $\mathcal{R}^{(k)}$ in these cases. Note that from Lemma 1 we see that for complex conjugate poles σ the factors $\rho_i^{(k)}$ in $\mathcal{R}^{(k)}$ are complex conjugate, too. Putting the terms with real coefficients first we thus have

$$\begin{aligned}
(-1)^k \cdot \mathcal{R}^{(k)}(t) &= \sum_{i=1}^{s'} \frac{\rho_i^{(k)}\omega_i}{t - \sigma_i} + \sum_{i=s'+1}^{s''} \left(\frac{\rho_i^{(k)}\omega_i}{t - \sigma_i} + \frac{\overline{\rho_i^{(k)}\omega_i}}{t - \overline{\sigma}_i} \right) \\
&= \sum_{i=1}^{s'} \frac{\rho_i^{(k)}\omega_i}{t - \sigma_i} + \sum_{i=s'+1}^{s''} \frac{\gamma_i^{(k)} t + \delta_i^{(k)}}{(t - \eta_i)^2 + \mu_i} \quad \text{with} \\
&\quad \gamma_i^{(k)} = 2\mathrm{Re}(\rho_i^{(k)}\omega_i), \delta_i^{(k)} = -2\mathrm{Re}(\rho_i^{(k)}\omega_i\overline{\sigma}_i), \\
&\quad \eta_i = \mathrm{Re}(\sigma_i), \mu_i = |\sigma_i|^2 - (\mathrm{Re}(\sigma_i))^2 > 0,
\end{aligned} \tag{8}$$

where the second line represents the real partial fraction expansion of $\mathcal{R}^{(k)}$. Note that we have $s'' = s'$ if the complex and the real partial fraction expansion coincide.

We now proceed by deriving bounds for the error using the rational functions $\mathcal{R}^{(k)}$. From standard norm estimates and using $\|v^{(k)}\| = 1$ we get from (6)

$$\|g(A)v - z^{(k)}\| \le \|\mathcal{R}^{(k)}(A)\|$$

and, in case that $\mathcal{R}^{(k)}(A)$ is non-singular,

$$\|g(A)v - z^{(k)}\| \ge \|(\mathcal{R}^{(k)}(A))^{-1}\|^{-1}.$$

Since we assume A to be Hermitian and since $\mathcal{R}^{(k)}$ is real, the matrix $\mathcal{R}^{(k)}(A)$ is Hermitian, too, so that

$$\|\mathcal{R}^{(k)}(A)\| = \max\{|\mu| : \mu \in \mathrm{spec}(\mathcal{R}^{(k)}(A))\} = \max\{|\mathcal{R}^{(k)}(\lambda)| : \lambda \in \mathrm{spec}(A)\}$$

and

$$\|\mathcal{R}^{(k)}(A)^{-1}\|^{-1} = \min\{|\mu| : \mu \in \mathrm{spec}(\mathcal{R}^{(k)}(A))\} = \min\{|\mathcal{R}^{(k)}(\lambda)| : \lambda \in \mathrm{spec}(A)\}.$$

In general, the quantities on the right hand side cannot be computed because the spectrum of A is not known. However, assuming that we know bounds ℓ_1, ℓ_2 such that $\operatorname{spec}(A) \subset [\ell_1, \ell_2]$ we can use the maximum and minimum over the whole interval, a computable quantity, to get bounds for the error. We summarize this next.

Theorem 1. *For $k = 1, 2, \ldots$ define*

$$\varepsilon^{(k)} = \min\{|\mathcal{R}^{(k)}(\lambda)| : \lambda \in [\ell_1, \ell_2]\}, \quad \mathcal{E}^{(k)} = \max\{|\mathcal{R}^{(k)}(\lambda)| : \lambda \in [\ell_1, \ell_2]\}. \tag{9}$$

Then $\varepsilon^{(k)} \leq \|g(A)v - z^{(k)}\| \leq \mathcal{E}^{(k)}$.

To solve the global optimization problems defining $\varepsilon^{(k)}$ and $\mathcal{E}^{(k)}$ we suggest to use a simple branch and bound method based on interval arithmetic which will be described in detail in the following section.

4 A branch and bound method based on interval arithmetic

We start by introducing some additional notation. (Compact) intervals on the real line are denote in boldface, as $\mathbf{x} = [\underline{\mathbf{x}}, \overline{\mathbf{x}}]$. The midpoint $(\underline{\mathbf{x}} + \overline{\mathbf{x}})/2$ of the interval $\mathbf{x}$ is denoted as $\operatorname{mid}(\mathbf{x})$, and $\operatorname{diam}(\mathbf{x}) = \overline{\mathbf{x}} - \underline{\mathbf{x}}$ is its diameter. The arithmetic operations $+, -, *, /$ on intervals are defined in a set theoretic manner as usually; their result is thus again a compact interval (see, e.g. [1], [14], [18]). The absolute value $|\mathbf{x}|$ is defined as the range of $|\cdot|$ over the interval. As with the arithmetic operations, it can be computed just from the endpoints of $\mathbf{x}$, since

$$|\mathbf{x}| = \begin{cases} [0, \max\{|\underline{\mathbf{x}}|, |\overline{\mathbf{x}}|\}] & \text{if } 0 \in \mathbf{x} \\ [\min\{|\underline{\mathbf{x}}|, |\overline{\mathbf{x}}|\}, \max\{|\underline{\mathbf{x}}|, |\overline{\mathbf{x}}|\}] & \text{otherwise} \end{cases}.$$

With these definitions, given the real partial fraction expansion of the rational function $\mathcal{R}^{(k)}$ from (8) and an interval $\mathbf{x} \subset \mathbb{R}$ which does not contain any real pole σ_i of $\mathcal{R}^{(k)}$, the *interval arithmetic evaluation* of $|\mathcal{R}^{(k)}|$ is defined as

$$|\mathcal{R}^{(k)}(\mathbf{x})| = \left| \sum_{i=1}^{s'} \frac{\rho_i^{(k)} \omega_i}{\mathbf{x} - \sigma_i} + \sum_{i=s'+1}^{s''} \frac{\gamma_i^{(k)} \mathbf{x} + \delta_i^{(k)}}{(\mathbf{x} - \eta_i)^2 + \mu_i} \right|.$$

By the inclusion property of interval arithmetic, the interval $|\mathcal{R}^{(k)}(\mathbf{x})|$ contains the range of $|\mathcal{R}^{(k)}|$ over the interval $\mathbf{x}$ which we denote by $\operatorname{Range}(|\mathcal{R}^{(k)}|, \mathbf{x})$. Since $|\mathcal{R}^{(k)}|$ satisfies a Lipschitz condition, we have that the difference $\operatorname{diam}(|\mathcal{R}^{(k)}(\mathbf{x})|) - \operatorname{diam}(\operatorname{Range}(|\mathcal{R}^{(k)}|, \mathbf{x}))$ tends to zero when $\operatorname{diam}(\mathbf{x})$ tends to zero; see, e.g., [1] or [18]. Note also that interval arithmetic evaluations are inclusion isotone, i.e. $\mathbf{y} \subset \mathbf{x} \Rightarrow |\mathcal{R}^{(k)}(\mathbf{y})| \subseteq |\mathcal{R}^{(k)}(\mathbf{y})|$.

For simplicity, we now use the generic notation f for the function $|\mathcal{R}^{(k)}|$. Given that we have an interval arithmetic evaluation of f at hand, we use a simple standard branch-and-bound strategy to obtain the global maximum, see

[9], [14] ,[22]. It relies on three ideas. The first is that if we keep on subdividing into ever smaller intervals, the interval arithmetic evaluations will tend towards the range of f. The second is that a global maximizer cannot lie in a subinterval $\mathbf{x}$ of $[\ell_1, \ell_2]$ for which $\overline{f(\mathbf{x})}$, the right end point of the interval arithmetic evaluation of f at $\mathbf{x}$, is less than the largest known value of f. The third is that lower and upper bounds for the global maximum of f may be easily updated by using computed function values and the right end points of the interval arithmetic evaluations, respectively. Technically, we will maintain a set of subintervals as a heap H containing pairs $(\mathbf{x}, \tilde{f})$. Here, each $\mathbf{x}$ is a subinterval of the initial interval $[\ell_1, \ell_2]$ and $\tilde{f} = \overline{f(\mathbf{x})}$, thus representing an upper bound for the maximum of f over that interval. The heap H is ordered with respect to the key $\tilde{f}$, so that if we retrieve the topmost element from the heap we always get the one with the largest value of $\tilde{f}$.

We keep track of two values f^* and $\hat{f}$ representing the best known values for which $\hat{f} \leq \mathcal{E} \leq f^*$, where $\mathcal{E} = \max_{x \in [\ell_1, \ell_2]} f(x)$. In each step we remove the entry $(\mathbf{x}, \tilde{f})$ with largest $\tilde{f}$ from H. The value of f^* is updated to be $\tilde{f}$. Then we bisect $\mathbf{x}$ into two intervals $\mathbf{x}_1$ and $\mathbf{x}_2$ and update $\hat{f}$ using the values of f at the midpoints of $\mathbf{x}_1$ and $\mathbf{x}_2$. We also compute $f(\mathbf{x}_i)$ giving us $\tilde{f}$ for both intervals $\mathbf{x}_i$. Only if $\tilde{f}$ is larger than $\hat{f}$ will the corresponding pair be inserted into the heap H. This is because otherwise $\mathbf{x}_i$ does not contain a global maximizer, and, by inclusion isotonicity, any subinterval of $\mathbf{x}_i$ will not contribute to further improve $\hat{f}$ or f^*.

We stop the bisection process once the difference between f^* and $\hat{f}$ is small enough. The following algorithm Maximize gives the details.

```
Algorithm MAXIMIZE
  Input: expression for function $f : [\ell_1, \ell_2] \subset \mathbb{R} \to \mathbb{R}_0^+$, relative accuracy $\alpha$
  Output: upper bound $f^*$ for global maximum $\mathcal{E}$
  $\mathbf{x} = [\ell_1, \ell_2]$, $\mathbf{f} = f(\mathbf{x})$, $\tilde{f} = \overline{\mathbf{f}}$
  insert $([\ell_1, \ell_2], \tilde{f})$ into empty heap $H$
  $f^* = \tilde{f}$, $\hat{f} = f(\mathrm{mid}(\mathbf{x}))$
  while $|(f^* - \hat{f})/f^*| > \alpha$ do
    remove top element $(\mathbf{x}, \tilde{f})$ from heap $H$          {has largest $\tilde{f}$}
    $f^* = \tilde{f}$                                   {improved upper bound for maximum}
    bisect $\mathbf{x} = \mathbf{x}_1 \cup \mathbf{x}_2$
    for $i = 1, 2$ do
      $\mathbf{f} = f(\mathbf{x}_i)$, $\tilde{f} = \overline{\mathbf{f}}$
      if $\tilde{f} > \hat{f}$ then {$\mathbf{x}_i$ may contain maximizer}
        insert $(\mathbf{x}_i, \tilde{f})$ in heap $H$
        $\hat{f} = \max\{\hat{f}, f(\mathrm{mid}(\mathbf{x}_i))\}$          {update largest function value}
      end if
    end for
  end while
```

Upon termination, this algorithm will have determined f^* as an upper bound for $\mathcal{E}$ with relative accuracy α, since $|(f^* - \hat{f})/f^*| \leq \alpha$ and $\hat{f} \leq \mathcal{E} \leq f^*$ imply $|(f^* - \mathcal{E})/f^*| \leq \alpha$.

Algorithm Maximize can be modified in a straightforward manner to deliver a lower bound for the minimum of f over the interval $[\ell_1, \ell_2]$.

5 The Lanczos algorithm with error bounds

We are now in a position to describe the Lanczos method to approximate $g(A)v$ in full detail, including the convergence test based on the error bound from Theorem 1, and its computation using the global optimization algorithm just described. In exact arithmetic this algorithm is guaranteed to yield an approximation for $g(A)b$ with the chosen accuracy. In floating point arithmetic no such guarantee can be given since the crucial relation

$$(A - \sigma_i I)x_i^{(k)} = (-1)^k \rho_i^{(k)} v^{(k)}$$

will not be fulfilled exactly with the computed quantities. Nevertheless, we consider our approach to be highly useful also in the floating point context since it produces a cheaply computable and, as numerical experiments will show, quite accurate stopping criterion.

Algorithm pfe-Lanczos.

```
Choose tol, maxit                                                {for stopping test}
$\beta = 0, v_0 = b, v_{-1} = 0, \rho_i^{(0)} = 1$ for $i = 1, \dots, s$
for $k = 1, \dots,$maxit do {iteration}
    $q = Av_{k-1} - \beta v_{k-2}, \;\; \alpha = v_{k-1}^* q, \;\; t_{k,k} = \alpha$     {Lanczos coeff's and vectors}
    $\widetilde{v} = q - \alpha v_{k-1}$
    $\beta = (\widetilde{v}^* \widetilde{v})^{1/2}, \;\; v_k = \widetilde{v}/\beta, \;\; t_{k+1,k} = \beta, \;\; t_{k,k+1} = \beta$
    $y_i = (T_k - \sigma_i I_k)^{-1} e_1, \;\; i = 1, \dots, s$                 {get projected solutions}
    $\rho_i^{(k)} = e_k^{\mathsf{T}} y_i t_{k+1,k}, \quad i = 1, \dots, s$                  {factors for residuals}
    Compute upper bound $\left(\mathcal{E}^{(k)}\right)^*$ for $\mathcal{E}^{(k)}$ with algorithm Maximize
                                                          {bounds from Theorem 1}
    if $\left(\mathcal{E}^{(k)}\right)^* <$ tol then {iteration converged}
        $z_k = \sum_{i=1}^{s} \omega_i y_i, \;\; x_k = \sum_{i=0}^{k-1} (z_k)_{i+1} v_i,$ stop         {approximate solution}
    end if
end for
```

We remark that underflow may occur for some of the numbers $\rho_i^{(k)}$ if the convergence for the corresponding systems is much faster than for others. It is therefore reasonable to incorporate a strategy to remove 'converged' systems from further computation and to set $\rho_i^{(k)} = 0$ for all subsequent iterations, see [6].

As already mentioned, in the presence of two complex conjugate poles, all quantities to be computed for the two poles are just complex conjugates of each other. Therefore, these computations have to be done for one of the poles only.

Finally, let us mention that there exists an alternative implementation of the Lanczos method which applies the conjugate gradient method to all s shifts simultaneously. Its advantage over Algorithm PFE-LANCZOS is that it requires storage only proportional to s, the number of poles, but not to the number of iterations performed. We refer to [6] for details.

6 Numerical experiments

In this section we report on the results of our numerical experiments. They were all programmed in Matlab with interval arithmetic provided through the Intlab toolbox; see [23]. In all experiments the accuracy parameter α in MAXIMIZE was taken to be 0.1.

Example 1. We consider the Zolotarev rational function approximation to the inverse square root function on a positive interval $[a_1, a_2]$, namely $\widetilde{g}(A)b \approx A^{-1/2}v$. We refer to [20, Chapter 4] for details on the Zolotarev approximation, which is a rational function with all simple poles lying on the negative real axis. We took $[a_1, a_2] = [1, 1000]$ and used the Zolotarev approximation with $s = 12$ poles. The matrix A was taken to be a 200×200 diagonal matrix A with diagonal entries equispaced in the interval $[1, 1000]$ so that $\ell_1 = 1, \ell_2 = 1000$; v was taken as the normalized vector of all ones. With these parameters, the accuracy of the Zolotarev approximation turns out to be of the order of 10^{-7}.

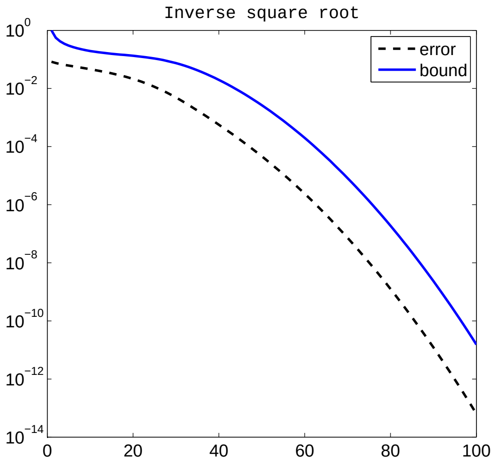

Fig. 1. Error and error bound for the inverse square root

Numerical results are given in Figure 1. The dotted curve represents the norm of the error. This error can be computed easily for this example, since, A

being diagonal, the exact value of $g(A)v$ can be computed directly. The solid line represents the error bound. We see that the error bound very well captures the convergence behaviour of the Lanczos method, while being roughly two orders of magnitude too pessimistic during the whole iteration process. We also see that by basing a stopping criterion upon the error bound bound we will perform about 10 'unnecessary' iterations for which the true error is already as small as desired while the error bound has yet to catch up.

Example 2. Our second experiment is with data stemming from an application in Quantum Chromodynamics; see [2]. We consider the approximation of $\mathrm{sign}(Q)b$, where Q is the Hermitian Wilson fermion matrix which is highly indefinite. We took $Q = P(I - \frac{4}{3}\kappa_c D)$, where $\kappa_c = 0.15717$, the configuration matrix D is available in the QCD collection of the matrix market [17] (matrix `conf5.4-00l8x8-2000.mtx`), while P is the so-called γ_5-matrix, a permutation matrix which symmetrizes D. The dimension of Q is approximately 50 000 and b is a random vector.

We first compute two numbers $0 < a_1 < a_2$ such that $\mathrm{spec}(Q) \subset [-a_2, -a_1] \cup [a_1, a_2]$. We then approximate $\mathrm{sign}(t)$ on $[-a_2, -a_1] \cup [a_1, a_2]$ using the Zolotarev rational approximation Z for the inverse square root on $[a_1^2, a_2^2]$. To be specific, we approximate $\mathrm{sign}(Q)b$ as $Z(Q^2) \cdot Qv$. Note that this means that we perform the Lanczos process using the matrix Q^2, not Q. Let us mention that $g(t) = Z(t^2) \cdot t$ is an ℓ_∞ best approximation to the sign function on $[-a_2, -a_1] \cup [a_1, a_2]$; see [20]. The number of poles s was chosen such that the ℓ_∞-error was less than 10^{-7}, that is $s = 11$. To speed up computation, it pays off to compute q eigenvalues of Q which are smallest in modulus, $\lambda_1, \dots, \lambda_q$, say, beforehand using a Lanczos procedure for Q^2. Denoting by $\mathit{\Pi}$ the orthogonal projector along the space spanned by the corresponding eigenvectors $w_i, i = 1, \dots, q$, we then work with the matrix $\mathit{\Pi} Q \mathit{\Pi}$ and the vector $\mathit{\Pi} b$. In this manner, we effectively shrink the eigenvalue intervals for Q, so that we need fewer poles for an accurate Zolotarev approximation and, in addition, the linear systems to be solved converge more rapidly. The vector $\mathrm{sign}(Q)b$ can be retrieved as $\mathrm{sign}(\mathit{\Pi} Q \mathit{\Pi})\mathit{\Pi} b + \mathrm{sign}(\mathrm{diag}(\lambda_1, \dots, \lambda_q)) \cdot (I - \mathit{\Pi})b$. In our computation we took $q = 30$, and the bounds ℓ_1, ℓ_2 for the the spectrum of the matrix $(\mathit{\Pi} Q \mathit{\Pi})^2$ were taken to be its smallest and largest nonzero eigenvalue, respectively.

Figure 2 shows the convergence curve and the error bound. Note that this time we compare with an 'exact' solution that has been computed beforehand by the Lanczos method. We see that the bounds for the error nicely reproduce the convergence behaviour and that they are about one order of magnitude larger than the true error.

Example 3. In this example we consider the Chebyshev rational approximation $g(A)b$ to the exponential function $\exp(-A)b$. The coefficients of the two polynomials of the same degree appearing in g have been tabulated in [5] for several different degrees. It is known that the error associated with this approximation is $\max_{t>0} |\exp(-t) - g(t)| = \mathcal{O}(10^{-s})$, where s is the degree of the polynomials in the rational function. In this case, the poles σ_i and the coefficients ω_i in the partial fraction expansion are complex, therefore pairing of the conjugate complex

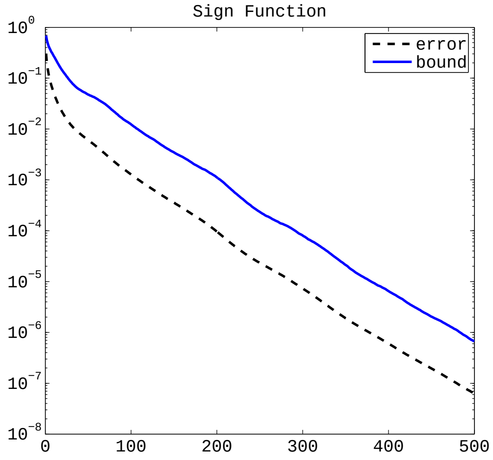

Fig. 2. Error and error bound for the sign function of the QCD matrix

terms in the partial fraction expansion should be carried out when computing the error bound, as discussed in (8). For our example we took A to be the standard 5 point discretization of the two-dimensional Laplacian on an equidistant grid of size 41×41. This results in a matrix of dimension 1600×1600 with eigenvalues $41^2 \cdot (8 - 4\cos^2(\pi/41 \cdot i) - 4\cos^2(\pi/41 \cdot j)), i, j = 1, \ldots, 40$, so that we took $[\ell_1, \ell_2] = 41^2 \cdot [1, 8]$. The convergence history is given in the left plot of Figure 3. We note that there is an initial stagnation phase, where during the first 40 iterations almost no progress is made. This stagnation phase is reflected in the error bound. Throughout the whole iteration, the error bound is about three orders of magnitude larger than the true error. However, since convergence tends to be quite fast after the initial stagnation phase, we again do not perform prohibitively many additional iterations if we base our stopping criterion upon the upper bound.

To speed up the convergence of the Lanczos process when the number of iterations becomes excessive, acceleration procedures have been devised. Here we consider the Shift-and-Invert Lanczos (SI-Lanczos), as proposed in [12] and [16]. For a given real parameter $\mu > 0$, the procedure determines an approximation to $f(A)b$ within the Krylov subspace $\mathcal{K}_k((I - \mu A)^{-1}, b)$. This space is generated by a Lanczos recurrence, and requires a solve with $(I - \mu A)$ at each iteration. This procedure is therefore effective only if one can solve these systems efficiently, e.g. using a multigrid method or a sparse direct solver.

For f a rational function, SI-Lanczos corresponds to approximating each system solution $(A - \sigma_i I)^{-1} b$ in the partial fraction expansion by projecting the problem onto $\mathcal{K}_k((I - \mu A)^{-1}, b)$ and then imposing the Galerkin condition ([21, Proposition 3.1]). More precisely, let $\widehat{A} = (I - \mu A)^{-1}$, $\widehat{\sigma}_i = 1/(\sigma_i \mu - 1)$ and

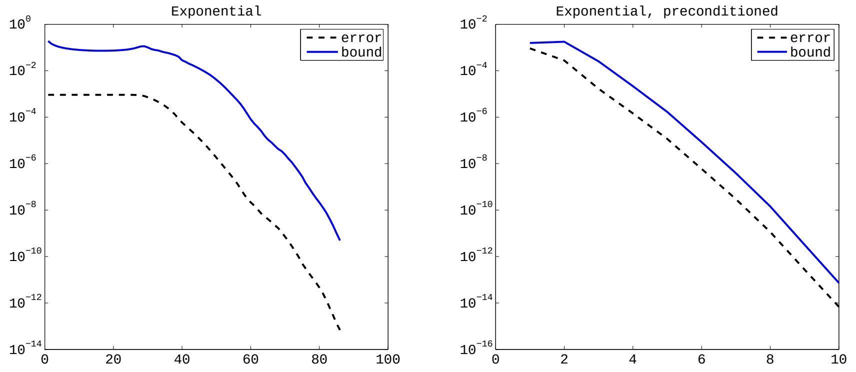

Fig. 3. Exponential function. Left: Standard Lanczos method. Right: SI-Lanczos method.

multiply both sides of $(A-\sigma_i I)x = b$ by $\widehat{A}$. Since $\widehat{A}\cdot(A-\sigma_i I) = \frac{1-\mu\sigma_i}{\mu}\left(\widehat{A}-\widehat{\sigma}_i I\right)$, for $i = 1, \ldots, s$, SI-Lanczos solves the systems

$$\left(\widehat{A}-\widehat{\sigma}_i I\right)\widehat{x} = \widehat{b}, \quad \text{with} \quad \widehat{x} = \frac{1-\mu\sigma_i}{\mu}x,\ \widehat{b} = \widehat{A}b. \tag{10}$$

The linear systems in (10) have precisely the same shifted structure as those in the previous sections. Let $\widehat{x}_i^{(k)}$ be the Galerkin solution to system i in $\mathcal{K}_k(\widehat{A}, \widehat{b})$, and let $x_i^{(k)} = \frac{\mu}{1-\mu\sigma_i}\widehat{x}_i^{(k)}$ be the corresponding approximate solution to the original system $(A-\sigma_i I)x = b$; see (10). Then

$$\begin{aligned} g(A)b - \sum_{i=1}^{s}\omega_i x_i^{(k)} &= \sum_{i=1}^{s}\omega_i(x_i - x_i^{(k)}) \\ &= \sum_{i=1}^{s}\frac{\mu\omega_i}{1-\mu\sigma_i}(\widehat{x}_i - \widehat{x}_i^{(k)}) \equiv \sum_{i=1}^{s}\widehat{\omega}_i(\widehat{x}_i - \widehat{x}_i^{(k)}). \end{aligned}$$

The procedure described in section 4 may thus be used to bound the error in the form given by the last expression. The right plot of Figure 3 contains the results for this approach, again for the 2D Laplacian on a 41×41 grid. Here we used $s = 14$ and $\mu = -1/\max_i |\sigma_i|$ (cf. [21]), so that now the interval $[\ell_1, \ell_2]$ is given by $\ell_1 = 1/(1-\mu a_2)$, $\ell_2 = 1/(1-\mu a_1)$, where $[a_1, a_2] = 41^2[1, 8]$ is the interval containing $\mathrm{spec}(A)$.

As expected, the number of iterations to achieve a given accuracy is reduced substantially. A stagnation phase is no longer present, and it is very remarkable that now the bound on the error has a tendency to get closer to the true error as the iteration proceeds.

References

1. G. Alefeld and J. Herzberger, *Introduction to Interval Computation*, Academic Press, 1983.
2. G. Arnold, N. Cundy, J. van den Eshof, A. Frommer, S. Krieg, T. Lippert, and K. Schäfer, *Numerical methods for the QCD overlap operator. II: Optimal Krylov subspace methods*, in QCD and Numerical Analysis III, A. Boriçi, A. Frommer, B. Joó, A. Kennedy, and B. Pendleton, eds., vol. 47 of Lecture Notes in Computational Science and Engineering, Springer, Berlin, 2005, pp. 153–167.
3. G. A. Baker and P. Graves-Morris, *Padé Approximants*, Encyclopedia of Mathematics and its applications, Cambridge University Press, Cambridge, 1996.
4. A. Borici, A. Frommer, B. Joó, A. Kennedy, and B. Pendleton, *QCD and numerical analysis III. Proceedings of the third international workshop on numerical analysis and lattice QCD, Edinburgh, UK, June 30 – July 4, 2003.*, Lecture Notes in Computational Science and Engineering 47. Berlin: Springer. xii, 201 p., 2005.
5. A. J. Carpenter, A. Ruttan, and R. S. Varga, *Extended numerical computations on the 1/9 conjecture in rational approximation theory*, in Rational Approximation and Interpolation, P. R. Graves-Morris, E. B. Saff, and R. S. Varga, eds., vol. 1105 of Lecture Notes in Mathematics, Springer-Verlag, Berlin, 1984, pp. 383–411.
6. 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.
7. V. Grimm and M. Hochbruck, *On the computation of certain trigonometric operator functions*, tech. rep., Mathematisches Institut, Heinrich-Heine-Universität Düsseldorf, 2008. `http://www.am.uni-duesseldorf.de/~marlis/publications/trigo.pdf`.
8. E. Hairer, C. Lubich, and G. Wanner, *Geometric numerical integration. Structure-preserving algorithms for ordinary differential equations*, vol. 31 of Springer Series in Computational Mathematics, Springer, Berlin, 2002.
9. E. R. Hansen and W. G. Walster, *Global Optimization Using Interval Analysis, 2nd Edition*, Marcel Dekker, New York, 2004.
10. N. J. Higham, *Matrix Functions – Theory and Applications*, SIAM, Philadelphia, USA, 2008.
11. M. Hochbruck and A. Ostermann, *Exponential Runge-Kutta methods for parabolic problems*, Applied Numer. Math., 53 (2005), pp. 323–339.
12. M. Hochbruck and J. van den Eshof, *Preconditioning Lanczos approximations to the matrix exponential*, SIAM J. Sci. Comput., 27 (2006), pp. 1438–1457.
13. R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge: Cambridge University Press, 1994.
14. R. B. Kearfott, *Rigorous Global Search: Continuous Problems*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.
15. The MathWorks, Inc., *MATLAB 7*, September 2004.
16. I. Moret and P. Novati, *RD-rational approximations of the matrix exponential*, BIT, Numerical Mathematics, 44 (2004), pp. 595–615.
17. National Institute of Standards and Technology, *Matrix market*.
18. A. Neumaier, *Interval Methods for Systems of Equations*, Cambridge University Press, 1990.

19. C. C. Paige, B. N. Parlett, and H. A. van der Vorst, *Approximate solutions and eigenvalue bounds from Krylov subspaces*, Numerical Linear Algebra with Applications, 2 (1995), pp. 115–134.
20. P. P. Petrushev and V. A. Popov, *Rational Approximation of Real Functions*, Cambridge University Press, Cambridge, 1987.
21. M. Popolizio and V. Simoncini, *Acceleration techniques for approximating the matrix exponential operator*, SIAM J. Matrix Analysis and Appl., (to appear).
22. H. Ratschek and J. Rokne, *New Computer Methods for Global Optimization*, Ellis Horwood, Chichester, New York, 1988.
23. S. M. Rump, *INTLAB – INTerval LABoratory*, in Developments in Reliabale Computing, T. Csendes, ed., Kluwer, Dordrecht, 1999, pp. 77–104.
24. 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.