



Approximating the extreme Ritz values and upper bounds for the A -norm of the error in CG

G rard Meurant¹ · Petr Tich ²

Received: 4 October 2018 / Accepted: 12 November 2018 / Published online: 20 November 2018
  Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

In practical conjugate gradient (CG) computations, it is important to monitor the quality of the approximate solution to $Ax = b$ so that the CG algorithm can be stopped when the required accuracy is reached. The relevant convergence characteristics, like the A -norm of the error or the normwise backward error, cannot be easily computed. However, they can be estimated. Such estimates often depend on approximations of the smallest or largest eigenvalue of A . In the paper, we introduce a new upper bound for the A -norm of the error, which is closely related to the Gauss-Radau upper bound, and discuss the problem of choosing the parameter μ which should represent a lower bound for the smallest eigenvalue of A . The new bound has several practical advantages, the most important one is that it can be used as an approximation to the A -norm of the error even if μ is not exactly a lower bound for the smallest eigenvalue of A . In this case, μ can be chosen, e.g., as the smallest Ritz value or its approximation. We also describe a very cheap algorithm, based on the incremental norm estimation technique, which allows to estimate the smallest and largest Ritz values during the CG computations. An improvement of the accuracy of these estimates of extreme Ritz values is possible, at the cost of storing the CG coefficients and solving a linear system with a tridiagonal matrix at each CG iteration. Finally, we discuss how to cheaply approximate the normwise backward error. The numerical experiments demonstrate the efficiency of the estimates of the extreme Ritz values, and show their practical use in error estimation in CG.

Keywords Conjugate gradients · Error norm estimation · Approximation of Ritz values · Incremental norm estimator

Mathematics Subject Classification (2010) 65F10 · 65F15 · 65F35

This work was supported by the project 17-04150J of the Grant Agency of the Czech Republic.

✉ Petr Tich 
petr.tichy@mff.cuni.cz

Extended author information available on the last page of the article.

1 Introduction

The (preconditioned) conjugate gradient ((P)CG) algorithm by Hestenes and Stiefel [18] is now considered as the iterative method of choice for solving linear systems $Ax = b$ with a real symmetric positive definite matrix A . An important question to solve practical problems is to know when to stop the iterations. Stopping criteria are application dependent. Some of them are based on the norm of the residual vector $r_k = b - Ax_k$ (where x_k is the approximate solution at iteration k), which is available in CG. For example, in optimization algorithms, $-r_k$ plays often the role of the gradient, and it can then be natural to stop CG based on $\|r_k\|$. However, in other applications, the use of $\|r_k\|$ as a measure of the quality of the approximation x_k can be misleading, as it was already mentioned in [18]. Moreover, in many cases, the residual norm is oscillating making the use of the stopping criteria based on $\|r_k\|$ more problematic.

In many applications, a natural stopping criterion could be based on the A -norm of the error

$$\|x - x_k\|_A = ((x - x_k)^T A (x - x_k))^{1/2}.$$

Mathematically, CG minimizes this quantity at each iteration k (see [18]). In some linear systems arising from engineering problems, the A -norm of the error corresponds to the energy norm and thus has a physical meaning. Of course, in real-world problems, the error and its norm are unknown. Therefore, this has led to some research works for finding approximations or even lower and upper bounds for the A -norm of the error. It turns out that the CG A -norm of the error is linked to a Riemann-Stieltjes integral for a discrete measure involving the distribution of the eigenvalues of A . Inspired by this connection already mentioned by Hestenes and Stiefel [18, p. 428], research on this topic was started by Gene Golub in the 1970s and continued throughout the years with several collaborators (e.g., G. Dahlquist, S. Eisenstat, S. Nash, B. Fischer, G. Meurant, Z. Strakoš). The main idea is to approximate the Riemann-Stieltjes integral by Gauss or Gauss-Radau quadrature rules. Since, in this case, the sign of the remainders of the quadrature rules are known, in theory, this gives lower and upper bounds for the A -norm of the error. These bounds can be used to design more reliable stopping criteria than just using the relative norm of the residual (for details on these techniques, see [4, 5, 9–11, 13]). This research was summarized in [12] and [22]. More recently, some simpler and improved formulas for the computation of the bounds on the A -norm of the error were provided in [23].

The techniques used in [10, 13, 29, 30] to compute lower or upper bounds use a positive integer d which is called the delay, in such a way that, at CG iteration $k + d$, an estimate of the A -norm of the error at iteration k is obtained. The larger the delay is, the better are the bounds at iteration k . However, even when using these techniques, the situation is still not completely satisfactory. Obtaining an upper bound with the Gauss-Radau quadrature rule needs to have a prescribed parameter which should represent a lower bound for the smallest eigenvalue of the (preconditioned) system matrix. This may not be readily available to the user. Moreover, some numerical examples have shown that, even if we have a good lower bound for the

smallest eigenvalue, the quality of the Gauss-Radau upper bound may deteriorate when the A -norm of the error becomes small. Sometimes, it is also useful to compute an approximation of the matrix 2-norm if the user wants to compute an estimate of the normwise backward error (see [25, 28]), or to approximate the ultimate level of accuracy, or the condition number of the (preconditioned) system matrix.

The goal of this paper is to discuss and address these issues to obtain cheap approximations to the smallest and largest eigenvalues of the (preconditioned) system matrix during the CG computations, and to use them in estimating convergence characteristics like the A -norm of the error or the normwise backward error. In particular, we introduce a new upper bound for the A -norm of the error which is less sensitive to the choice of the approximation to the smallest eigenvalue and suggest an approximation of this upper bound which does not require any a priori information about the smallest eigenvalue.

The paper is organized as follows. In Section 2, we recall the Lanczos and CG algorithms as well as some relations which show the links between CG and Gauss quadrature. Section 3 is concerned with the Gauss-Radau upper bound and the derivation of a new upper bound. In Section 4, we present a numerical example that shows the troubles that may happen with the Gauss-Radau upper bounds, and a possible potential of the new upper bound which is not sensitive to the choice of the approximation to the smallest eigenvalue. In Section 5, we address the problem of computing estimates of the smallest and largest eigenvalues of A . This is done by using incremental estimates of norms of bidiagonal matrices. These algorithms can be useful in a more general setting than computing bounds for the CG error norms. In Sections 6 and 7, these results are used to approximate the Gauss-Radau upper bound and the normwise backward error. Section 8 illustrates numerically the quality of approximations to the smallest and largest eigenvalues, and their use in approximating the normwise backward error and the A -norm of the error. Finally, in Section 9, we give some conclusions and perspectives.

2 The Lanczos and CG algorithms

Given a starting vector $v \in \mathbb{R}^N$ and a symmetric matrix $A \in \mathbb{R}^{N \times N}$, one can consider a sequence of nested subspaces

$$\mathcal{K}_k(A, v) \equiv \text{span} \{v, Av, \dots, A^{k-1}v\}, \quad k = 1, 2, \dots,$$

called Krylov subspaces. The dimension of these subspaces is increasing up to an index $n \leq N$ called the *grade of v with respect to A* , at which the maximal dimension is attained, and $\mathcal{K}_n(A, v)$ is invariant under multiplication with A . Assuming that $k < n$, the Lanczos algorithm (Algorithm 1) computes an orthonormal basis v_1, \dots, v_{k+1} of the Krylov subspace $\mathcal{K}_{k+1}(A, v)$. The basis vectors v_j of unit norm satisfy the matrix relation

$$AV_k = V_k T_k + \tilde{\beta}_k v_{k+1} e_k^T$$

Algorithm 1 Lanczos algorithm

```

input  $A, v$ 
 $\tilde{\beta}_0 = 0, v_0 = 0$ 
 $v_1 = v / \|v\|$ 
for  $k = 1, \dots$  do
     $w = Av_k - \tilde{\beta}_{k-1}v_{k-1}$ 
     $\tilde{\alpha}_k = v_k^T w$ 
     $w = w - \tilde{\alpha}_k v_k$ 
     $\tilde{\beta}_k = \|w\|$ 
     $v_{k+1} = w / \tilde{\beta}_k$ 
end for

```

where $V_k = [v_1 \cdots v_k]$, e_k denotes the k th column of the identity matrix, and

$$T_k = \begin{bmatrix} \tilde{\alpha}_1 & \tilde{\beta}_1 & & \\ \tilde{\beta}_1 & \ddots & \ddots & \\ & \ddots & \ddots & \tilde{\beta}_{k-1} \\ & & \tilde{\beta}_{k-1} & \tilde{\alpha}_k \end{bmatrix}$$

is the $k \times k$ symmetric tridiagonal matrix of the recurrence coefficients computed in Algorithm 1. The coefficients $\tilde{\beta}_j$ being positive, T_k is a Jacobi matrix. The Lanczos algorithm works for any symmetric matrix, but if A is positive definite, then T_k is positive definite as well.

When solving a system of linear equations $Ax = b$ with a real symmetric positive definite matrix A , the CG method (Algorithm 2) can be used. Mathematically, the CG iterates x_k minimize the A -norm of the error over the manifold $x_0 + \mathcal{K}_k(A, r_0)$,

$$\|x - x_k\|_A = \min_{y \in x_0 + \mathcal{K}_k(A, r_0)} \|x - y\|_A,$$

Algorithm 2 Conjugate gradients

```

input  $A, b, x_0$ 
 $r_0 = b - Ax_0$ 
 $p_0 = r_0$ 
for  $k = 1, \dots$  until convergence do
     $\gamma_{k-1} = \frac{r_{k-1}^T r_{k-1}}{p_{k-1}^T A p_{k-1}}$ 
     $x_k = x_{k-1} + \gamma_{k-1} p_{k-1}$ 
     $r_k = r_{k-1} - \gamma_{k-1} A p_{k-1}$ 
     $\delta_k = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}$ 
     $p_k = r_k + \delta_k p_{k-1}$ 
end for

```

and the residual vectors $r_k = b - Ax_k$ are proportional to the Lanczos vectors v_j ,

$$v_{j+1} = (-1)^j \frac{r_j}{\|r_j\|}, \quad j = 0, \dots, k.$$

Thanks to this close relationship between the CG and Lanczos algorithms it can be shown (see, for instance [22]) that the recurrence coefficients computed in both algorithms are connected via

$$\tilde{\beta}_k = \frac{\sqrt{\delta_k}}{\gamma_{k-1}}, \quad \tilde{\alpha}_k = \frac{1}{\gamma_{k-1}} + \frac{\delta_{k-1}}{\gamma_{k-2}}, \quad \delta_0 = 0, \quad \gamma_{-1} = 1. \quad (1)$$

Writing these formulas in matrix form, we get

$$T_k = L_k L_k^T, \quad L_k^T = \begin{bmatrix} \frac{1}{\sqrt{\gamma_0}} & \sqrt{\frac{\delta_1}{\gamma_0}} & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & \ddots & \sqrt{\frac{\delta_{k-1}}{\gamma_{k-2}}} \\ & & & & \frac{1}{\sqrt{\gamma_{k-1}}} \end{bmatrix}.$$

In other words, CG computes implicitly the Cholesky factorization of the Jacobi matrix T_k generated by the Lanczos algorithm. Hence, the eigenvalues of T_k (the so-called Ritz values) are equal to the squared singular values of the upper bidiagonal matrix L_k^T .

It is well known that the reduction of the squared A -norm of the error from iteration $k - 1$ to iteration k is given by $\gamma_{k-1} \|r_{k-1}\|^2$ (see [18, relation (6:1)]). As a consequence

$$\|x - x_0\|_A^2 = \sum_{j=0}^{k-1} \gamma_j \|r_j\|^2 + \|x - x_k\|_A^2. \quad (2)$$

The relation (2) represents the basis for the quadrature-based estimation of the A -norm of the error in the CG method [10, 11, 13, 23, 24, 29, 30]. In more details, let $A = Q\Lambda Q^T$ be the spectral decomposition of A , with $Q = [q_1, \dots, q_N]$ orthonormal and $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_N)$, the λ_i 's, $i = 1, \dots, N$ being the eigenvalues of A . For simplicity of notation, we assume that the eigenvalues of A are distinct and ordered as $\lambda_1 < \lambda_2 < \dots < \lambda_N$. Let us define the weights ω_i by

$$\omega_i \equiv \frac{(r_0, q_i)^2}{\|r_0\|^2} \quad \text{so that} \quad \sum_{i=1}^N \omega_i = 1, \quad (3)$$

and the (nondecreasing) stepwise constant distribution function $\omega(\lambda)$ with a finite number of points of increase $\lambda_1, \lambda_2, \dots, \lambda_N$,

$$\omega(\lambda) \equiv \begin{cases} 0 & \text{for } \lambda < \lambda_1, \\ \sum_{j=1}^i \omega_j & \text{for } \lambda_i \leq \lambda < \lambda_{i+1}, \quad 1 \leq i \leq N-1, \\ 1 & \text{for } \lambda_N \leq \lambda. \end{cases} \quad (4)$$

Having the distribution function $\omega(\lambda)$ and an interval $\langle \zeta, \xi \rangle$ such that $\zeta < \lambda_1 < \lambda_2 < \dots < \lambda_N < \xi$, for any continuous function f , one can define the Riemann-Stieltjes integral (see, for instance [12])

$$\int_{\zeta}^{\xi} f(\lambda) d\omega(\lambda) = \sum_{i=1}^N \omega_i f(\lambda_i). \quad (5)$$

For the integrated function defined as $f(\lambda) = \lambda^{-1}$, we obtain the integral representation of the squared initial A -norm of the error

$$\begin{aligned} \|x - x_0\|_A^2 &= r_0^T A^{-1} r_0 = (Q^T r_0)^T \Lambda^{-1} (Q^T r_0) \\ &= \|r_0\|^2 \sum_{j=1}^n \lambda_j^{-1} \omega_j = \|r_0\|^2 \int_{\zeta}^{\xi} \lambda^{-1} d\omega(\lambda). \end{aligned}$$

Finally, using the optimality of CG, it can be shown that the formula (2) represents the scaled k -point Gauss quadrature rule for approximating the Riemann-Stieltjes integral of the function $f(\lambda) = \lambda^{-1}$, with the scaled positive remainder $\|x - x_k\|_A^2$. The scaling factor is $\|r_0\|^{-2}$. Various modified quadrature rules can be used to obtain other approximations to the integral, possibly also with a negative remainder. Such rules usually require some a priori information about the spectrum of A (for a summary, see, e.g., the book [12]).

3 Quadrature-based bounds and a new upper bound

In this section, we concentrate on two simple upper bounds. To summarize some results of [10, 13, 29], and [23] related to the Gauss and Gauss-Radau quadrature bounds for the A -norm of the error in CG, it has been shown that

$$\gamma_k \|r_k\|^2 < \|x - x_k\|_A^2 < \gamma_k^{(\mu)} \|r_k\|^2 \quad (6)$$

where

$$\gamma_{k+1}^{(\mu)} = \frac{(\gamma_k^{(\mu)} - \gamma_k)}{\mu (\gamma_k^{(\mu)} - \gamma_k) + \delta_{k+1}}, \quad \gamma_0^{(\mu)} = \frac{1}{\mu}, \quad (7)$$

$k < n - 1$, and μ such that $0 < \mu \leq \lambda_{\min}$. Note that in the special case $k = n - 1$ since $\|x - x_n\|_A^2 = 0$, we get $\|x - x_{n-1}\|_A^2 = \gamma_{n-1} \|r_{n-1}\|^2$. If the initial residual r_0 has a nontrivial component in the eigenvector corresponding to λ_{\min} , then λ_{\min} is also an eigenvalue of T_n . If in addition μ is chosen such that $\mu = \lambda_{\min}$, then $\gamma_{n-1} = \gamma_{n-1}^{(\mu)}$ and the second strict inequality in (6) changes to equality.

The simple updating formula (7) was first presented in [23]. Following the idea of [13] and [29], we can improve the lower and upper bounds in (6) by considering quadrature rules (2) at iterations k and $k + d$ for some integer $d > 0$ which is called the delay. Then, we get the formula

$$\|x - x_k\|_A^2 = \sum_{j=k}^{k+d-1} \gamma_j \|r_j\|^2 + \|x - x_{k+d}\|_A^2, \quad (8)$$

and one can bound the error norm at iteration $k + d$ using (6) to obtain

$$\sum_{j=k}^{k+d-1} \gamma_j \|r_j\|^2 + \gamma_{k+d} \|r_{k+d}\|^2 < \|x - x_k\|_A^2 \quad (9)$$

and

$$\|x - x_k\|_A^2 < \sum_{j=k}^{k+d-1} \gamma_j \|r_j\|^2 + \gamma_{k+d}^{(\mu)} \|r_{k+d}\|^2. \quad (10)$$

Note that (9) and (10) give a lower bound and an upper bound for the A -norm of the error at iteration k when CG is already at iteration $k + d$ whence (6) provides lower and upper bounds when CG is at iteration k . In [29], it has been shown that the identity (8) holds (up to some small inaccuracies) also for numerically computed quantities in finite precision arithmetic, until the A -norm of the error reaches its ultimate level of accuracy. So, it can be used safely for estimating the A -norm of the actual error.

Mathematically, we will derive another upper bound for the squared A -norm of the error, which is closely related to the Gauss-Radau upper bound. This bound depends on the ratio

$$\phi_k \equiv \frac{\|r_k\|^2}{\|p_k\|^2},$$

which can be updated using a simple recurrence relation. In particular, using $p_k = r_k + \delta_k p_{k-1}$ and the orthogonality between r_k and p_{k-1} (local orthogonality), we obtain

$$\|p_k\|^2 = \|r_k\|^2 \left(1 + \delta_k \frac{\|p_{k-1}\|^2}{\|r_{k-1}\|^2} \right),$$

and, therefore,

$$\phi_k = \frac{\phi_{k-1}}{\phi_{k-1} + \delta_k}, \quad \phi_0 = 1. \quad (11)$$

Hence, ϕ_k can be updated cheaply without computing the norm of p_k which is not readily available in CG. From (11) and by induction, it follows that

$$\phi_k^{-1} = 1 + \frac{\|r_k\|^2}{\|r_{k-1}\|^2} \phi_{k-1}^{-1} = \|r_k\|^2 \sum_{j=0}^k \|r_j\|^{-2}$$

and hence

$$\|r_k\|^2 \phi_k = \left(\sum_{j=0}^k \|r_j\|^{-2} \right)^{-1}; \quad (12)$$

(see also [18, Theorem 5:3]). Note that mathematically, the quantity (12) can be interpreted as the norm of the residual vector determined by the minimal residual method (see, e.g., [8, Theorem 3.5]). In finite precision arithmetic, the quantity (12) cannot be, in general, interpreted as the norm of the residual vector generated by some

minimal residual method. We remark that the quantity ϕ_k appears also as a coefficient in strategies for residual smoothing [16, 17]. In particular, one can compute the smoothed residual r_k^S and the corresponding approximation x_k^S using the recurrences

$$r_k^S = (1 - \phi_k) r_{k-1}^S + \phi_k r_k, \quad x_k^S = (1 - \phi_k) x_{k-1}^S + \phi_k x_k.$$

The new upper bound is as follows.

Theorem 1 *Let $0 < \mu \leq \lambda_{\min}$ be given. The approximations x_k , $k < n$, generated by the CG method satisfy*

$$\|x - x_k\|_A^2 < \frac{\|r_k\|^2}{\mu} \frac{\|r_k\|^2}{\|p_k\|^2}, \quad (13)$$

and the bound is decreasing with increasing k .

Proof Based on (6), it is sufficient to show that $\mu\gamma_k^{(\mu)} \leq \phi_k$. We will prove it by induction. The inequality holds for $k = 0$. Using the induction hypothesis, (7), and (11) we obtain, for $k < n - 1$,

$$\mu\gamma_{k+1}^{(\mu)} = \frac{\mu(\gamma_k^{(\mu)} - \gamma_k)}{\mu(\gamma_k^{(\mu)} - \gamma_k) + \delta_{k+1}} < \frac{\mu\gamma_k^{(\mu)}}{\mu\gamma_k^{(\mu)} + \delta_{k+1}} \leq \frac{\phi_k}{\phi_k + \delta_{k+1}} = \phi_{k+1}.$$

Recall that $\gamma_k^{(\mu)} - \gamma_k$ is positive because of (6). Finally, using (12), the bound (13) is monotonically decreasing with increasing k . \square

The tightness of the bound (13) can further be improved when using a delay d , similarly as in (10). First, the proof of the previous theorem also shows that the Gauss-Radau upper bound presented in (6) can be bounded from above by

$$\gamma_k^{(\mu)} \|r_k\|^2 < \frac{\|r_k\|^2}{\mu} \frac{\|r_k\|^2}{\|p_k\|^2}. \quad (14)$$

Second, combining (10) and (14), we can get an improved upper bound

$$\|x - x_k\|_A^2 < \sum_{j=k}^{k+d-1} \gamma_j \|r_j\|^2 + \frac{\|r_{k+d}\|^2}{\mu} \frac{\|r_{k+d}\|^2}{\|p_{k+d}\|^2}. \quad (15)$$

In practical computations, the parameter μ has to be determined. This represents a nontrivial task.

4 A numerical example: the choice of μ

As an example that can demonstrate the difficulties to compute accurate upper bounds for the A -norm of the error, we consider the matrix bcsstk01 from the set BCSSTRUC1 in the Harwell-Boeing collection, which can be obtained from the

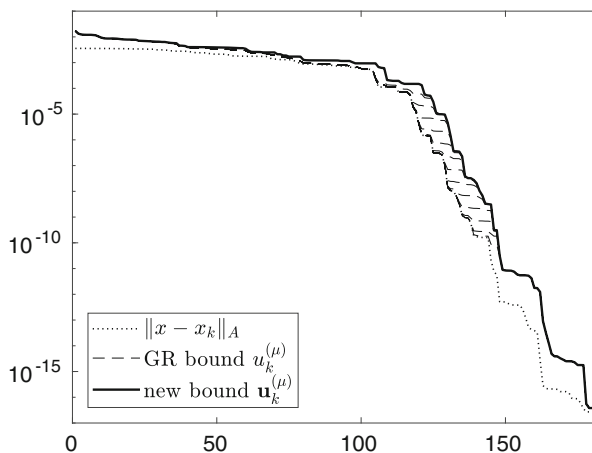


Fig. 1 bcsstk01, $u_k^{(\mu)}$ and $\mathbf{u}_k^{(\mu)}$, $\mu = \lambda_{\min}/(1 + 10^{-m})$, $m = 2, \dots, 14$

Matrix Market¹ or from the SuiteSparse Matrix Collection.² It is a small stiffness matrix of order 48 arising from dynamic analysis in structural engineering with 400 nonzero entries. Its condition number is $\kappa(A) = 8.8234 \times 10^5$. The smallest eigenvalue $\lambda_{\min}(A) = 3.417267562666500 \times 10^3$ was computed in extended precision and rounded to double precision. The right-hand side b has been chosen such that b has equal components in the eigenvector basis, and such that $\|b\| = 1$.

The linear system $Ax = b$ is difficult to solve with CG without a preconditioner. We have to perform around 180 iterations to reach the maximum attainable accuracy when the matrix is only of order 48. There is a long phase of quasi-stagnation of the A -norm of the error that last almost 100 iterations as one can see in Fig. 1. Denote

$$u_k^{(\mu)} \equiv \sqrt{\gamma_k^{(\mu)}} \|r_k\|, \quad \mathbf{u}_k^{(\mu)} \equiv \sqrt{\frac{\phi_k}{\mu}} \|r_k\| \quad (16)$$

the bounds which correspond to (6) and (13) (without any delay $d = 0$).

Figure 1 displays the A -norm of the error (dotted curve), the bounds $u_k^{(\mu)}$ for different values of μ equal to $\lambda_{\min}/(1 + 10^{-m})$, $m = 2, \dots, 14$ (dashed curves), and the new upper bounds $\mathbf{u}_k^{(\mu)}$ (thick solid curves). The closer μ is to λ_{\min} the better is the upper bound $u_k^{(\mu)}$ of the A -norm of the error. However, below a level of approximately 10^{-8} , all the values of μ in our experiment give visually the same upper bound $u_k^{(\mu)}$ which is not very close to the A -norm of the error. We can also observe that the new upper bound $\mathbf{u}_k^{(\mu)}$ is insensitive to the choice of μ and gives an envelope of the Gauss-Radau upper bounds $u_k^{(\mu)}$.

Figure 2 shows the “upper bounds” $u_k^{(\mu)}$ for values of μ which are larger than but close to λ_{\min} ; $\mu = \lambda_{\min}/(1 - 10^{-m})$, $m = 2, 4, 6, \dots, 14$. We use quotes since, as

¹<http://math.nist.gov/MatrixMarket>

²<https://sparse.tamu.edu/>

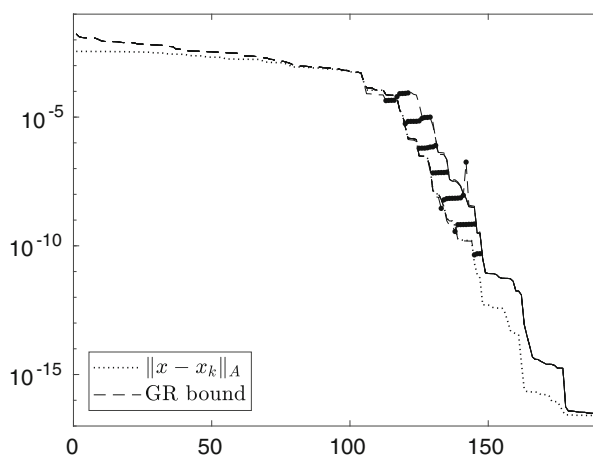


Fig. 2 bcsttk01, $|\gamma_k^{(\mu)}|^{1/2} \|r_k\|$, $\mu = \lambda_{\min}/(1 - 10^{-m})$, $m = 2, 4, 6, \dots, 14$ (dashed curves). Large dots emphasize this quantity if $\gamma_k^{(\mu)} < 0$

one can see, we do not obtain an upper bound in general, even though we are close to λ_{\min} . If μ is chosen to be larger than λ_{\min} , then, at some point, the coefficient $\gamma_k^{(\mu)}$ can even be negative. In such cases, we use $|\gamma_k^{(\mu)}|$ and emphasize the corresponding value by a dot. In Fig. 2, we do not plot the new bound $u_k^{(\mu)}$. However, from its definition and the assumption that $\mu \approx \lambda_{\min}$, it follows that $u_k^{(\mu)}$ will stay visually the same as in Fig. 1. Note that the dashed curves in Fig. 2 are overlapping in the final stage of convergence and look like a solid curve.

In summary, the node μ should satisfy $\mu \leq \lambda_{\min}$, and, simultaneously, it should closely approximate λ_{\min} ; otherwise, the Gauss-Radau upper bound $u_k^{(\mu)}$ would be a poor approximation of the A -norm of the error. If the smallest eigenvalue is known in advance, then the bound $u_k^{(\mu)}$ can give very good results until some level of accuracy of the error norm (in our case 10^{-8}) is reached. Below this level, the bounds $u_k^{(\mu)}$ and $u_k^{(\mu)}$ visually coincide, and are far away from the A -norm of the norm.

If the parameter μ has to be determined, possibly in some adaptive way, then we can expect troubles. First, one cannot hope in general to get a very accurate approximation of the smallest eigenvalue without too much work. Second, there is usually no guarantee that the condition $\mu \leq \lambda_{\min}$ is satisfied. Typically, the best we can get from the Lanczos process are the Ritz values (eigenvalues of T_k) which can approximate the eigenvalues of A . However, Ritz values provide only upper bounds on λ_{\min} , and some heuristics (e.g., multiplication by a safety constant) have to be used to obtain μ with the desired properties. As we have seen in the numerical example, the value $u_k^{(\mu)}$ can be very sensitive to small perturbations of μ . Then, using a heuristic can strongly influence the approximation properties of $u_k^{(\mu)}$ and cause numerical troubles in computation of $u_k^{(\mu)}$ if $\mu > \lambda_{\min}$. On the other hand, the new bound $u_k^{(\mu)}$ can be computed without any troubles also for $\mu > \lambda_{\min}$. If in addition $\mu \approx \lambda_{\min}$, then either $u_k^{(\mu)}$ represents an upper bound, or, it is an approximation of the A -norm of the error. For example, an approximation μ_k of the smallest Ritz value can be used as a heuristic

to approximate the upper bound $\mathbf{u}_k^{(\mu)}$ using $\mathbf{u}_k^{(\mu_k)}$. Since the upper bound $\mathbf{u}_k^{(\mu)}$ is not too sensitive to the choice of μ , one can expect that the approximation $\mathbf{u}_k^{(\mu_k)}$ can give reasonable results even if μ_k is only a moderate approximation to the smallest Ritz value, and the smallest Ritz value is a poor estimate of the smallest eigenvalue. This fact will be demonstrated later in the numerical experiments of Section 8.

5 Approximating the extreme Ritz values

In this section, we develop efficient algorithms for the incremental approximation of the smallest and largest Ritz values. This information can be used not only in the error approximation techniques based on various modified quadrature rules (see, e.g., [10, 11, 23]), but also to approximate the 2-norm of A or the condition number of A . Note that an approximation of $\|A\|$ is needed in estimating the maximum attainable accuracy (see [15]) or in the computation of the normwise backward error (see [28]).

As already mentioned, Jacobi matrices T_k and the lower bidiagonal matrices L_k which appear in CG are related through $T_k = L_k L_k^T$. In particular, it holds that

$$\lambda_{\max}(T_k) = \|L_k\|^2, \quad \lambda_{\min}(T_k) = \|L_k^{-1}\|^{-2}. \quad (17)$$

Hence, one can approximate the extreme eigenvalues of T_k using incremental norm estimation applied to the upper triangular matrices L_k^T and L_k^{-T} . Although we are mainly motivated by the approximation of the extreme Ritz values in CG, we consider the problem of incremental norm estimation of bidiagonal matrices and their inverses by itself, since it can be useful also in other algorithms involving bidiagonal matrices.

5.1 The eigenvalues and eigenvectors of a 2×2 symmetric matrix

An important ingredient of incremental norm estimation is the fact that the eigenvalues and eigenvectors of a 2×2 symmetric matrix are known explicitly. Consider a matrix of the form

$$\begin{bmatrix} \rho & \sigma \\ \sigma & \tau \end{bmatrix}. \quad (18)$$

The two eigenvalues of (18) are given by

$$\lambda_+ = \frac{1}{2}(\rho + \tau + \chi), \quad \lambda_- = \frac{1}{2}(\rho + \tau - \chi)$$

where

$$\chi^2 = (\rho - \tau)^2 + 4\sigma^2. \quad (19)$$

If $\sigma \neq 0$, the matrix of unnormalized eigenvectors is given by

$$\begin{bmatrix} \rho - \tau + \chi & \rho - \tau - \chi \\ 2\sigma & 2\sigma \end{bmatrix}.$$

For more details see [6, p.306], [12, p.166].

5.2 Incremental estimation of the norms of upper triangular matrices

To approximate the maximum singular value of an upper triangular matrix, we use an incremental estimator proposed in [6]. The algorithm is based on incremental improvement of an approximation of the right singular vector that corresponds to the maximum singular value. In [7], it has been shown that this technique tends to be superior, with respect to approximating maximum singular values, to the original incremental technique proposed in [3]. In the following, we recall the basic idea of the incremental norm estimation and reformulate the algorithm slightly so that it can be efficiently applied to upper bidiagonal matrices and their inverses.

Let $R \in \mathbb{R}^{k \times k}$ be an upper triangular matrix and let z be its approximate (or exact) maximum right singular vector. Let

$$\widehat{R} = \begin{bmatrix} R & v \\ & \eta \end{bmatrix}, \quad v \in \mathbb{R}^k, \quad \eta \in \mathbb{R}, \quad (20)$$

and consider the new approximate maximum right singular vector in the form

$$\widehat{z} = \begin{bmatrix} sz \\ c \end{bmatrix}, \quad (21)$$

where $s^2 + c^2 = 1$. The parameters s and c are chosen such that the norm of the vector $\widehat{R}\widehat{z}$ is maximal. It holds that

$$\|\widehat{R}\widehat{z}\|^2 = \begin{bmatrix} s \\ c \end{bmatrix}^T \begin{bmatrix} \rho & \sigma \\ \sigma & \tau \end{bmatrix} \begin{bmatrix} s \\ c \end{bmatrix}$$

where

$$\rho = \|Rz\|^2, \quad \sigma = v^T Rz, \quad \tau = v^T v + \eta^2.$$

Hence, to maximize $\|\widehat{R}\widehat{z}\|^2$, we need to determine the maximum eigenvalue of the symmetric 2×2 matrix (18), and the corresponding eigenvector. Using the previous results

$$\begin{bmatrix} s \\ c \end{bmatrix} = \frac{u}{\|u\|}, \quad u = \begin{bmatrix} \rho - \tau + \chi \\ 2\sigma \end{bmatrix}, \quad (22)$$

and

$$\lambda_+ = \frac{\rho + \tau + \chi}{2}, \quad \chi^2 = (\rho - \tau)^2 + 4\sigma^2.$$

Note that if $\sigma = 0$, the formula for the eigenvector that corresponds to λ_+ is still valid. Next, it holds that

$$\|u\|^2 = 2(\chi^2 + (\rho - \tau)\chi),$$

and, therefore, from (19),

$$c^2 = \frac{2\sigma^2}{\chi^2 + (\rho - \tau)\chi} = \frac{1}{2} \frac{\chi^2 - (\rho - \tau)^2}{\chi^2 + (\rho - \tau)\chi} = \frac{1}{2} \left(1 - \frac{\rho - \tau}{\chi} \right).$$

We can also express $\|\widehat{R}\widehat{z}\|^2$ in a more convenient form

$$\|\widehat{R}\widehat{z}\|^2 = \frac{\rho + \tau + \chi}{2} = \rho + \frac{\chi}{2} \left(1 - \frac{\rho - \tau}{\chi} \right) = \rho + \chi c^2.$$

To compute \widehat{z} , we still need to determine the signs of s and c . From (22), it follows that $s \geq 0$ and c has the same sign as σ . Therefore,

$$s = \sqrt{1 - c^2}, \quad c = |c| \text{sign}(\sigma).$$

Using the subscript k , we can formulate Algorithm 3 for the incremental norm estimation of

$$R_{k+1} = \begin{bmatrix} R_k & v_k \\ & \eta_k \end{bmatrix}, \quad v_k \in \mathbb{R}^k, \quad \eta_k \in \mathbb{R}, \quad (23)$$

where R_k in Algorithm 3 is a principal submatrix of R_{k+1} .

Algorithm 3 Incremental estimation of $\|R_k\|^2$

```

input matrices  $R_k$ 
 $z_1 = 1$ ,
for  $k = 1, \dots$  do

    % ... Compute the entries of the  $2 \times 2$  matrix.

     $\rho_k = \|R_k z_k\|^2$ ,  $\sigma_k = v_k^T R_k z_k$ ,  $\tau_k = v_k^T v_k + \eta_k^2$ 

    % ... Compute the new estimate  $\rho_{k+1}$ .

     $\chi_k^2 = (\rho_k - \tau_k)^2 + 4\sigma_k^2$ ,  $c_k^2 = \frac{1}{2} \left( 1 - \frac{\rho_k - \tau_k}{\chi_k} \right)$ ,  $\rho_{k+1} = \rho_k + \chi_k c_k^2$ 

    % ... If required, compute  $z_{k+1}$ .

     $s_k = \sqrt{1 - c_k^2}$ ,  $c_k = |c_k| \text{sign}(\sigma_k)$ ,  $z_{k+1} = \begin{bmatrix} s_k z_k \\ c_k \end{bmatrix}$ 

end for
```

Note that if we start the algorithm with $z_1 = 1$, then $\rho_1 = r_{1,1}^2$, and ρ_2 is equal to $\|R_2\|^2$. In more details, it holds that

$$\begin{aligned} \rho_2 &= \rho_1 + \chi_1 c_1^2 = \frac{1}{2} (\chi_1 + \rho_1 + \tau_1) \\ &= \frac{1}{2} \left(r_{1,1}^2 + r_{2,2}^2 + r_{1,2}^2 + \sqrt{\left(r_{1,1}^2 - r_{2,2}^2 - r_{1,2}^2 \right)^2 + 4r_{1,1}^2 r_{1,2}^2} \right) = \|R_2^T R_2\|. \end{aligned}$$

As we will see in the following, if R_k is upper bidiagonal, it is possible to incrementally estimate $\|R_k\|$ and $\|R_k^{-1}\|$ in a very efficient way, without storing the coefficients of the matrix R_k and even without storing the approximate right singular vectors z_k . In particular, we will be able to find simple updating formulas for σ_k and τ_k which are then used in the updating formula for ρ_{k+1} .

5.3 Specialization to upper bidiagonal matrices

Consider a bidiagonal matrix B_k ,

$$B_k = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ & \alpha_2 & \beta_2 & & \\ & & \ddots & \ddots & \\ & & & \ddots & \beta_{k-1} \\ & & & & \alpha_k \end{bmatrix}. \quad (24)$$

Having relation (23) in mind and taking $R_k = B_k$, the vector v_k and the entry η_k in the last column of B_{k+1} are given by $v_k = \beta_k e_k$, $\eta_k = \alpha_{k+1}$, where $e_k = [0, \dots, 0, 1]^T$ is the k th column of the $k \times k$ identity matrix. Hence,

$$\rho_k = \|B_k z_k\|^2, \quad \sigma_k = \alpha_k \beta_k e_k^T z_k, \quad \tau_k = \beta_k^2 + \alpha_{k+1}^2.$$

Note that the last entry $e_k^T z_k$ of the vector z_k is given by c_{k-1} (see (21)), and, therefore, $\sigma_k = \alpha_k \beta_k c_{k-1}$. Using the previous results, we are now able to update the entries ρ_k , σ_k and τ_k without storing the vector z_k (see Algorithm 4).

Algorithm 4 Incremental estimation of $\|B_k\|^2$

input entries α_k and β_k of upper bidiagonal matrices

$\rho_1 = \alpha_1^2$, $\rho_1^{\max} = \rho_1$, $c_0 = 1$,

for $k = 1, \dots$ **do**

$\sigma_k^2 = \alpha_k^2 \beta_k^2 c_{k-1}^2$, $\tau_k = \beta_k^2 + \alpha_{k+1}^2$

$\chi_k^2 = (\rho_k - \tau_k)^2 + 4\sigma_k^2$

$c_k^2 = \frac{1}{2} \left(1 - \frac{\rho_k - \tau_k}{\chi_k} \right)$

$\rho_{k+1} = \rho_k + \chi_k c_k^2$

$\rho_{k+1}^{\max} = \rho_{k+1}$

end for

In some cases, a better accuracy of the approximations to norms of matrices is needed. To improve the accuracy, we need to store B_k and z_k so that we can run Algorithm 3, and construct the approximate maximum right singular vector

$$z_{k+1} = \begin{bmatrix} s_k z_k \\ c_k \end{bmatrix} \quad (25)$$

of B_{k+1} . The vector z_{k+1} can also be seen as an approximate eigenvector of $B_{k+1}^T B_{k+1}$ corresponding to the approximate maximum eigenvalue ρ_{k+1} . Hence, one can improve the vector z_{k+1} using one shifted inverse iteration applied to the matrix $B_{k+1}^T B_{k+1}$, where ρ_{k+1} is used as a shift (see, e.g., [14, Section 7.6]).

In detail, having the LDL^T factorization of the tridiagonal matrix $B_{k+1}^T B_{k+1}$, we can easily compute the LDL^T factorization of the matrix $B_{k+1}^T B_{k+1} - \rho_{k+1} I$ using

the `dstqds` algorithm by Parlett and Dhillon [27]. The last factorization can be used to perform one inverse iteration by solving the system

$$\left(B_{k+1}^T B_{k+1} - \rho_{k+1} I\right) y = z_{k+1}.$$

Finally, we can consider the vector $\widehat{z}_{k+1} \equiv y/\|y\|$ and the scalar $\widehat{\rho}_{k+1} \equiv \|B_{k+1} \widehat{z}_{k+1}\|^2$ to be new approximations to the maximum right singular vector and to the squared norm of B_{k+1} , and $\widehat{\rho}_{k+1}^{\max} \equiv \widehat{\rho}_{k+1}$ to be an improved estimate of the largest eigenvalue of $B_{k+1}^T B_{k+1}$.

5.4 Inversions of nonsingular upper bidiagonal matrices

Consider a nonsingular bidiagonal matrix B_k of the form (24), $\alpha_i \neq 0$. It is well known that the last column w_{k+1} of the matrix B_{k+1}^{-1} can be expressed in the explicit form

$$w_{k+1} = \frac{1}{\alpha_{k+1}} \left[(-1)^k \prod_{i=1}^k \frac{\beta_i}{\alpha_i}, \dots, \frac{\beta_{k-1}}{\alpha_{k-1}} \frac{\beta_k}{\alpha_k}, -\frac{\beta_k}{\alpha_k}, 1 \right]^T.$$

Hence,

$$B_{k+1}^{-1} = \begin{bmatrix} B_k^{-1} & -w_k \frac{\beta_k}{\alpha_{k+1}} \\ & \frac{1}{\alpha_{k+1}} \end{bmatrix}, \quad w_{k+1} = \frac{1}{\alpha_{k+1}} \begin{bmatrix} -w_k \beta_k \\ 1 \end{bmatrix}, \quad (26)$$

where w_k is the last column of the matrix B_k^{-1} . We now specialize the idea of the incremental norm estimation presented in Section 5.2 to the case of matrices B_k^{-1} , that is,

$$R_k = B_k^{-1}, \quad v_k = -w_k \frac{\beta_k}{\alpha_{k+1}}, \quad \eta_k = \frac{1}{\alpha_{k+1}}.$$

First, let us find updating formulas for $\|w_{k+1}\|^2$ and $B_k^{-T} w_k$. From (26), it follows that

$$\|w_{k+1}\|^2 = \frac{1}{\alpha_{k+1}^2} \left(\beta_k^2 \|w_k\|^2 + 1 \right), \quad (27)$$

and

$$B_k^{-T} w_k = \begin{bmatrix} B_{k-1}^{-T} \\ -w_{k-1}^T \frac{\beta_{k-1}}{\alpha_k} \frac{1}{\alpha_k} \end{bmatrix} \begin{bmatrix} -w_{k-1} \frac{\beta_{k-1}}{\alpha_k} \\ \frac{1}{\alpha_k} \end{bmatrix} = \begin{bmatrix} -\frac{\beta_{k-1}}{\alpha_k} \left(B_{k-1}^{-T} w_{k-1} \right) \\ \|w_k\|^2 \end{bmatrix}. \quad (28)$$

Using the formulas (27) and (28), we are now able to update the entries σ_k and τ_k which are needed in the process of the incremental norm estimation (see Section 5.2). For τ_k we get

$$\tau_k = \|w_{k+1}\|^2 = \frac{1}{\alpha_{k+1}^2} \left(\beta_k^2 \|w_k\|^2 + 1 \right) = \frac{1}{\alpha_{k+1}^2} \left(\beta_k^2 \tau_{k-1} + 1 \right),$$

and for σ_k ,

$$\begin{aligned}\sigma_k &= v_k^T R_k z_k = -\frac{\beta_k}{\alpha_{k+1}} z_k B_k^{-T} w_k \\ &= -\frac{\beta_k}{\alpha_{k+1}} \begin{bmatrix} s_{k-1} z_{k-1} \\ c_{k-1} \end{bmatrix}^T \begin{bmatrix} -\frac{\beta_{k-1}}{\alpha_k} (B_{k-1}^{-T} w_{k-1}) \\ \|w_k\|^2 \end{bmatrix} \\ &= -\frac{\beta_k}{\alpha_{k+1}} \left(s_{k-1} \left[-\frac{\beta_{k-1}}{\alpha_k} z_{k-1}^T B_{k-1}^{-T} w_{k-1} \right] + c_{k-1} \|w_k\|^2 \right) \\ &= -\frac{\beta_k}{\alpha_{k+1}} (s_{k-1} \sigma_{k-1} + c_{k-1} \tau_{k-1}).\end{aligned}$$

The initial values

$$\rho_1 = \frac{1}{\alpha_1^2}, \quad \tau_0 = \frac{1}{\alpha_1^2}, \quad \sigma_0 = 0, \quad s_0 = 0, \quad c_0 = 1,$$

lead to the 2×2 matrix

$$\begin{bmatrix} \rho_1 & \sigma_1 \\ \sigma_1 & \tau_1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\alpha_1^2} & -\frac{\beta_1}{\alpha_2 \alpha_1^2} \\ -\frac{\beta_1}{\alpha_2 \alpha_1^2} & \frac{1}{\alpha_2^2} + \left(\frac{\beta_1}{\alpha_2 \alpha_1} \right)^2 \end{bmatrix} = B_2^{-T} B_2^{-1},$$

so that $\rho_2 = \|B_2^{-1}\|^2$. The results are summarized in Algorithm 5.

Algorithm 5 Incremental estimation of $\|B_k^{-1}\|^2$

input entries α_k and β_k of upper bidiagonal matrices
 $\rho_1 = \alpha_1^{-2}$, $\rho_1^{\min} = \alpha_1^2$, $\tau_0 = \rho_1$, $\sigma_0 = 0$, $s_0 = 0$, $c_0 = 1$
for $k = 1, \dots$ **do**
 $\sigma_k = -\frac{\beta_k}{\alpha_{k+1}} (s_{k-1} \sigma_{k-1} + c_{k-1} \tau_{k-1})$,
 $\tau_k = \frac{1}{\alpha_{k+1}^2} (\beta_k^2 \tau_{k-1} + 1)$
 $\chi_k^2 = (\rho_k - \tau_k)^2 + 4\sigma_k^2$
 $c_k^2 = \frac{1}{2} \left(1 - \frac{\rho_k - \tau_k}{\chi_k} \right)$
 $\rho_{k+1} = \rho_k + \chi_k c_k^2$
 $s_k = \sqrt{1 - c_k^2}$, $c_k = |c_k| \operatorname{sign}(\sigma_k)$
 $\rho_{k+1}^{\min} = \rho_{k+1}^{-1}$
end for

Similarly as in the previous section, we can improve the accuracy of the approximations of norms of inverses of matrices by one shifted inverse iteration. To do so, we need to store B_k , z_k , and also the vector $B_k^{-T} w_k$ (to compute σ_k) which can be updated using the formula (28). Then, as in (25), we can construct the approximate maximum right singular vector z_{k+1} of B_{k+1}^{-1} . The vector z_{k+1} can be seen as an

approximate eigenvector of the matrix $B_{k+1}^{-T} B_{k+1}^{-1}$, or, as an approximate eigenvector of the matrix $B_{k+1} B_{k+1}^T$,

$$B_{k+1}^{-T} B_{k+1}^{-1} z_{k+1} \approx \rho_{k+1} z_{k+1}, \quad \rho_{k+1}^{-1} z_{k+1} \approx B_{k+1} B_{k+1}^T z_{k+1}.$$

The accuracy of the vector z_{k+1} can now be improved by one shifted inverse iteration applied to the matrix $B_{k+1} B_{k+1}^T$, where ρ_{k+1}^{-1} is used as a shift.

In detail, we can easily get the UDU^T factorization (U is upper bidiagonal) of the tridiagonal matrix $B_{k+1} B_{k+1}^T$. Using a straightforward modification of the `dstqds` algorithm, the UDU^T factorization of the matrix $B_{k+1} B_{k+1}^T - \rho_{k+1}^{-1} I$ can be computed and used to solve the system

$$(B_{k+1} B_{k+1}^T - \rho_{k+1}^{-1} I) y = z_{k+1}.$$

The modification of the `dstqds` algorithm consists in the unitary transformation of the problem for the UDU^T factorization to the problem with LDL^T factorization, using the backward identity matrix. Finally, one can consider the vector $\widehat{z}_{k+1} \equiv y/\|y\|$ and the scalar $\widehat{\rho}_{k+1} \equiv \|B_{k+1}^{-1} \widehat{z}_{k+1}\|^2$ to be new approximations to the maximum right singular vector and to $\|B_{k+1}^{-1}\|^2$, and $\widehat{\rho}_{k+1}^{\min} \equiv \widehat{\rho}_{k+1}^{-1}$ to be an improved estimate of the smallest eigenvalues of $B_{k+1}^T B_{k+1}$.

5.5 CG and approximations of the extreme Ritz values

The results of the previous sections can be applied to the upper bidiagonal matrices $B_k = L_k^T$ that are computed in CG, i.e.,

$$\alpha_j = \frac{1}{\sqrt{\gamma_{j-1}}}, \quad j = 1, \dots, k, \quad \beta_j = \sqrt{\frac{\delta_j}{\gamma_{j-1}}}, \quad j = 1, \dots, k-1,$$

to approximate the smallest and largest eigenvalues of T_k (see (17)). In particular, after substitution, we obtain in Algorithm 4,

$$\sigma_k^2 = \frac{\delta_k}{\gamma_{k-1}^2} c_{k-1}^2, \quad \tau_k = \frac{1}{\gamma_{j-1}} + \frac{\delta_k}{\gamma_{j-1}},$$

and in Algorithm 5,

$$\sigma_k = -\sqrt{\gamma_k \frac{\delta_k}{\gamma_{k-1}}} (s_{k-1} \sigma_{k-1} + c_{k-1} \tau_{k-1}), \quad \tau_k = \gamma_k \left(\delta_k \frac{\tau_{k-1}}{\gamma_{k-1}} + 1 \right). \quad (29)$$

Moreover, for τ_k in Algorithm 5, it holds that

$$\frac{\tau_k}{\gamma_k} = [1 + \delta_k(1 + \delta_{k-1}(1 + \dots + \delta_2(1 + \delta_1) \dots))] = \frac{\|p_k\|^2}{\|r_k\|^2}.$$

6 Approximation of the Gauss-Radau upper bound

The previous section provides a cheap tool to approximate the Gauss-Radau upper bound without having an a priori information about the smallest eigenvalue of the (preconditioned) system matrix. In particular, to approximate the Gauss-Radau upper bound, one can use the new upper bound (13). Instead of μ which should closely approximate the smallest eigenvalue from below, one can use the updated approximation $\mu_k \equiv \rho_k^{\min}$ to the smallest Ritz value (see Algorithm 5 and Section 5.5). Since the bound (13) is not sensitive to the choice of μ , the approximation of (13) which uses μ_k will be close to the bound (13) for $\mu = \lambda_{\min}$ whenever $\mu_k \approx \lambda_{\min}$. Moreover, as we have seen in Section 4, the bound (13) is often a good approximation to the Gauss-Radau upper bound, in particular if μ approximates the smallest eigenvalue only roughly, say to 1 or 2 valid digits. In summary, when we do not have an a priori information about the smallest eigenvalue of the (preconditioned) system matrix, we suggest to estimate the Gauss-Radau upper bound $u_k^{(\mu)}$ (see (16)), using an approximation

$$u_k^{(\mu_k)} = \frac{\|r_k\|}{\sqrt{\mu_k}} \frac{\|r_k\|}{\|p_k\|} = \sqrt{\frac{\phi_k}{\mu_k}} \|r_k\| \quad (30)$$

where $\mu_k = \rho_k^{\min}$ is updated at each iteration as in Algorithm 5, with σ_k and τ_k computed directly from the CG coefficients using (29). The algorithm for updating μ_k starts with $\rho_1 = \gamma_0$, $\mu_1 = \gamma_0^{-1}$, $\tau_0 = \rho_1$, $\sigma_0 = 0$, $s_0 = 0$, $c_0 = 1$. Note that it does not make too much sense to use inverse iterations to improve the quality of the approximation of the smallest Ritz value. A more accurate approximation to the smallest Ritz value does not improve the approximation (30) significantly.

7 Approximation of the normwise backward error

In [1, 26], backward error perturbation theory was used to derive a family of stopping criteria for iterative methods. In particular, given \tilde{x} , one can ask what the norms are of the smallest perturbations ΔA of A and Δb of b measured in the relative sense such that the approximate solution \tilde{x} represents the exact solution of the perturbed system

$$(A + \Delta A) \tilde{x} = b + \Delta b.$$

In other words, we are interested in the quantity

$$\eta = \min \{ \delta : (A + \Delta A) \tilde{x} = b + \Delta b, \|\Delta A\| \leq \delta \|A\|, \|\Delta b\| \leq \delta \|b\| \}.$$

It was shown by Rigal and Gaches [28] that this quantity, called *the normwise backward error*, is given by

$$\eta = \frac{\|\tilde{r}\|}{\|A\| \|\tilde{x}\| + \|b\|}. \quad (31)$$

where $\tilde{r} = b - A\tilde{x}$. This approach can be generalized (see [1, 26]) in order to quantify levels of confidence in A and b . The normwise backward error is, as a base for stopping criteria, frequently recommended in the numerical analysis literature (see, e.g. [2, 19]).

When solving a linear system with CG, the norms of vectors $\tilde{x} = x_k$ and $\tilde{r} = r_k$ are easily computable, and $\|A\|$ can be approximated from below using Algorithm 4 (see also Section 5.5). Hence, we can efficiently compute an upper bound on the normwise backward error (31) in CG. In the following subsection, we show that if $x_0 = 0$, then $\|x_k\|$ can be approximated cheaply in an incremental way.

7.1 A cheap approximation of $\|x_k\|$ in CG

If $x_0 = 0$, then the CG approximate solution x_k can be expressed as

$$x_k = \|r_0\| V_k T_k^{-1} e_1, \quad \text{and} \quad \|x_k\|^2 = \|r_0\|^2 e_1^T T_k^{-1} V_k^T V_k T_k^{-1} e_1.$$

Using the *global orthogonality* among the Lanczos vectors, we obtain

$$\|x_k\|^2 = \|r_0\|^2 e_1^T T_k^{-2} e_1. \quad (32)$$

Note that in finite precision arithmetic, the orthogonality is usually quickly lost. However, we observed in numerical experiments (see Section 8) that despite the loss or orthogonality, the quantity

$$\xi_k \equiv \|r_0\|^2 e_1^T T_k^{-2} e_1 \quad (33)$$

still approximates $\|x_k\|^2$ very accurately. In the following lemma, we suggest an algorithm to efficiently compute ξ_k at a negligible cost.

Lemma 1 *With the notation introduced in Section 2, it holds that*

$$\xi_k = \sum_{j=0}^{k-1} \|r_j\|^{-2} \left(\sum_{i=j}^{k-1} \psi_i \right)^2, \quad \psi_i = \gamma_i \|r_i\|^2,$$

and ξ_{k+1} , $k = 0, 1, 2, \dots$, can be computed using the recurrences

$$\vartheta_{k+1} = \vartheta_k + \gamma_k \phi_k^{-1}, \quad (34)$$

$$\xi_{k+1} = \xi_k + \psi_k (\vartheta_{k+1} + \vartheta_k), \quad (35)$$

where $\vartheta_0 = 0$, $\xi_0 = 0$, and ϕ_k can be updated using (11).

Proof It holds that

$$\xi_k = \|r_0\|^2 e_1^T T_k^{-2} e_1 = \|\|r_0\| L_k^{-T} L_k^{-1} e_1\|^2 \equiv \|y\|^2$$

where $y = [y_1, \dots, y_k]^T$ solves the system $L_k^T L_k y = \|r_0\| e_1$. Using the bidiagonal structure of L_k , we get in a straightforward way that

$$y_j = (-1)^{j+1} \frac{1}{\|r_{j-1}\|} \left(\sum_{i=j-1}^{k-1} \psi_i \right), \quad j = 1, \dots, k,$$

and, therefore,

$$\xi_k = \|y\|^2 = \sum_{j=0}^{k-1} \frac{\left(\sum_{i=j}^{k-1} \psi_i \right)^2}{\|r_j\|^2}.$$

It remains to find a way how to compute ξ_k in an efficient way. In other words, knowing ξ_k and ψ_k , we would like to express ξ_{k+1} . It holds that

$$\begin{aligned}\xi_{k+1} &= \sum_{j=0}^k \frac{\left(\sum_{i=j}^k \psi_i\right)^2}{\|r_j\|^2} = \frac{\psi_k^2}{\|r_k\|^2} + \sum_{j=0}^{k-1} \frac{\psi_k^2}{\|r_j\|^2} + \sum_{j=0}^{k-1} \frac{2\psi_k \sum_{i=j}^{k-1} \psi_i}{\|r_j\|^2} + \xi_k \\ &= \psi_k \left(\sum_{j=0}^k \frac{\sum_{i=j}^k \psi_i}{\|r_j\|^2} + \sum_{j=0}^{k-1} \frac{\sum_{i=j}^{k-1} \psi_i}{\|r_j\|^2} \right) + \xi_k \\ &= \xi_k + \psi_k (\vartheta_{k+1} + \vartheta_k)\end{aligned}$$

where

$$\vartheta_k \equiv \sum_{j=0}^{k-1} \frac{\sum_{i=j}^{k-1} \psi_i}{\|r_j\|^2}.$$

Let us find an updating formula for ϑ_{k+1} . We have,

$$\begin{aligned}\vartheta_{k+1} &= \gamma_k + \sum_{j=0}^{k-1} \frac{\sum_{i=j}^{k-1} \psi_i + \psi_k}{\|r_j\|^2} = \gamma_k + \vartheta_k + \psi_k \sum_{j=0}^{k-1} \frac{1}{\|r_j\|^2} \\ &= \gamma_k + \vartheta_k + \gamma_k \frac{\|r_k\|^2}{\|r_{k-1}\|^2} \left(\|r_{k-1}\|^2 \sum_{j=0}^{k-1} \|r_j\|^{-2} \right) \\ &= \gamma_k + \vartheta_k + \gamma_k \delta_k \phi_{k-1}^{-1} \\ &= \vartheta_k + \gamma_k \phi_k^{-1};\end{aligned}$$

see (11) and (12). \square

Lemma 1 shows how to cheaply approximate $\|x_k\|$ in CG under the assumption $x_0 = 0$. If $x_0 \neq 0$, then $x_k - x_0 = \|r_0\| V_k T_k^{-1} e_1$ and ξ_k can be seen as an approximation to $\|x_k - x_0\|^2$. Considering preconditioned CG (PCG) (see Algorithm 6) with a symmetric and positive definite preconditioner M , quantities ξ_k (computed using (11) and (34)–(35) from the coefficients and residual norms which correspond to the preconditioned system) will approximate $\|x_k - x_0\|_M^2$. The norms $\|x_k\|_M$ of preconditioned approximations can be of interest when approximating the normwise backward error which corresponds to the preconditioned system.

7.2 Normwise backward error in PCG

Given a symmetric positive definite matrix $M = LL^T$, we can formally think about preconditioned CG (see Algorithm 6) as CG applied to the modified system

$$\underbrace{L^{-1}AL^{-T}}_{\hat{A}} \underbrace{L^T x}_{\hat{x}} = \underbrace{L^{-1}b}_{\hat{b}}. \quad (36)$$

Moreover, a change of variable is used to go back to the original variable x and the original residual r in such a way that the only preconditioning matrix which is involved is M or its inverse, and not L which may be unknown. Using the techniques

Algorithm 6 Preconditioned conjugate gradients (PCG)

input A, b, x_0, M ,
 $r_0 = b - Ax_0$, solve $Mz_0 = r_0$ to get $z_0, p_0 = z_0$
for $k = 1, \dots$ **until** convergence **do**
 $\hat{\gamma}_{k-1} = \frac{z_{k-1}^T r_{k-1}}{p_{k-1}^T A p_{k-1}}$
 $x_k = x_{k-1} + \hat{\gamma}_{k-1} p_{k-1}$
 $r_k = r_{k-1} - \hat{\gamma}_{k-1} A p_{k-1}$
 solve $Mz_k = r_k$ to get z_k
 $\hat{\delta}_k = \frac{z_k^T r_k}{z_{k-1}^T r_{k-1}}$
 $p_k = z_k + \hat{\delta}_k p_{k-1}$
end for

presented in Sections 5 and 7.1, we can approximate the normwise backward error for the preconditioned system (36),

$$\tilde{\eta} = \frac{\|\tilde{r}\|}{\|\hat{A}\| \|\tilde{x}\| + \|\hat{b}\|}, \quad (37)$$

where \tilde{x} is a given approximation and $\tilde{r} = \hat{b} - \hat{A}\tilde{x}$. In particular, in PCG, we are interested in $\tilde{x} = L^T x_k, \tilde{r} = L^{-1} r_k$, so that

$$\|\tilde{x}\|^2 = \|x_k\|_M^2, \quad \|\tilde{r}\|^2 = z_k^T r_k = \|r_k\|_{M^{-1}}^2, \quad \|\hat{b}\|^2 = \|b\|_{M^{-1}}^2. \quad (38)$$

The norm of the preconditioned matrix $\|\hat{A}\|$ can be approximated from the PCG coefficients $\hat{\gamma}_k$ and $\hat{\delta}_k$ using techniques developed in Section 5, and the norm of the preconditioned approximation $\|L^T x_k\| = (x_k^T M x_k)^{1/2} = \|x_k\|_M$ can be approximated using Lemma 1. The other quantities are available in PCG.

We know that \tilde{x} is the exact solution of a perturbed problem $(\hat{A} + \Delta\hat{A})\tilde{x} = \hat{b} + \Delta\hat{b}$, where the relative sizes of $\Delta\hat{A}$ and $\Delta\hat{b}$ are bounded by $\tilde{\eta}$. Hence, x_k is the exact solution of the perturbed system

$$(A + \hat{\Delta}_A)x_k = b + \hat{\Delta}_b, \quad \hat{\Delta}_A \equiv L(\Delta\hat{A})L^T, \quad \hat{\Delta}_b \equiv L(\Delta\hat{b}).$$

Since the relative sizes of $\Delta\hat{A}$ and $\Delta\hat{b}$ are bounded by $\tilde{\eta}$, it holds that $\|\hat{\Delta}_A\|/\|A\| \leq \kappa(M)\tilde{\eta}$ and $\|\hat{\Delta}_b\|/\|b\| \leq \kappa(M)^{1/2}\tilde{\eta}$.

Nevertheless, the question of which backward error makes more sense in a given problem remains. The quantity η in (31) tells us how well we have solved the original system whence $\tilde{\eta}$ in (37) tells us how well we have solved the preconditioned system. We did not find any discussion of this issue in the literature.

8 Numerical experiments

Numerical experiments are divided into two parts. In the first part, we demonstrate the quality of our estimates approximating the extreme Ritz values and the norms of approximate solutions during the CG computations. In the second part, we use these estimates to approximate characteristics of our interest, i.e., the Gauss-Radau upper

bound for the A -norm of the error and the normwise backward error. The experiments are performed in Matlab 9.2 (R2017a).

We consider four systems of linear equations. The first one with the system matrix `bcsstk01` has already been described in Section 4. For this system, the influence of finite precision arithmetic to CG computations is substantial; orthogonality is quickly lost and convergence is significantly delayed. Hence, one can test whether our techniques work also under these circumstances which are quite realistic during practical computations.

The second system arises after discretizing the diffusion equation

$$-\operatorname{div}(\lambda(x, y) \nabla u) = f \quad \text{in } \Omega = (0, 1)^2, \quad u|_{\partial\Omega} = 0,$$

with the diffusion coefficient

$$\lambda(x, y) = \frac{1}{2 + 1.8 \sin(10x)} \cdot \frac{1}{2 + 1.8 \sin(10y)}.$$

The PDE is discretized using standard finite differences with a five-point scheme on a 60×60 mesh so that the system matrix `Pb26` has the moderate dimension 3600 (for more details, see [22, Section 9.2, p. 313]). Note that $\operatorname{nnz}(A) = 17760$ and $\kappa(A) \approx 7.54 \times 10^4$. The right-hand side b is a random vector normalized to have a unit norm. The starting vector is $x_0 = 0$. In the experiments, the system is solved without preconditioning.

The third linear system `Pres_Poisson` from the SuiteSparse Matrix Collection arises in problems of computational fluid dynamics. The matrix size is $n = 14822$, $\operatorname{nnz}(A) = 715804$, $\kappa(A) \approx 2.04 \times 10^6$, the right-hand side b is provided with the matrix. The starting vector is $x_0 = 0$. We use incomplete Cholesky factorization with zero-fill as a preconditioner (see, e.g., [14, Section 11.5.8]). To compute this factorization, we use the Matlab command `ichol(A)`.

Finally, the last system matrix `s3dkt3m2` is of order $n = 90449$ and $\kappa(A) \approx 3.6 \times 10^{11}$. It can be downloaded from the CYLSHELL collection in the Matrix Market library, which contains matrices that represent low-order finite-element discretizations of a shell element test, the pinched cylinder. Only the last element of the right-hand side vector b is nonzero, which corresponds to the given physical problem (for more details, see [20] and the references therein). The factor L in the preconditioner $M = LL^T$ is determined by the incomplete Cholesky factorization with threshold dropping. To compute this factorization, we use the Matlab command `ichol` with parameters `type = 'ict'`, `droptol = 1e-5`, and with the global diagonal shift `diagcomp = 1e-2`. Note that here $\operatorname{nnz}(A) = 3686223$ and $\operatorname{nnz}(L) = 6541916$. When used in experiments, the smallest eigenvalue of the preconditioned matrix was computed as the smallest Ritz value at the iteration $k = 3500$ for which the ultimate level of accuracy of the A -norm of the error was already reached.

8.1 Approximations to the extreme Ritz values and to $\|x_k\|$

It is sometimes difficult to know beforehand good approximations of the smallest and largest eigenvalues of A . Since CG is equivalent to the Lanczos algorithm, estimates

of the smallest and largest eigenvalues can be computed during CG iterations via approximating the smallest and largest Ritz value. In Algorithms 4 and 5, and in Section 5.5, we formulated a very cheap way of approximating the extreme Ritz values at a negligible cost of a few scalar operations per iteration. Moreover, the estimates can be improved when updating the LDL^T factorization of the tridiagonal matrix T_k and performing one shifted inverse iteration (see Sections 5.3 and 5.4).

Note that an adaptive algorithm for approximating the smallest eigenvalue was also proposed in [21], with the aim to get the parameter μ for computing the Gauss-Radau bound. The user was required to provide an initial lower bound for $\lambda_{\min}(A)$. Then, during the CG iterations the smallest Ritz values were computed using a fixed number of inverse iterations. When the smallest Ritz value was considered to be converged, the value of μ was changed to the converged value. However, this required solving several tridiagonal linear systems at every CG iteration and the size of these linear systems was increasing with the CG iterations. Therefore, we can do now something better with our new cheap estimates, as well as with the improved estimates which require solving of just one linear system per iteration.

Let us first describe the meaning of curves in Figs. 3, 4, 5, and 6. The left and right parts of the figures correspond to approximations of the largest and smallest eigenvalue, respectively. Denote by $\theta_1^{(k)}, \dots, \theta_k^{(k)}$ the eigenvalues of T_k , i.e., the Ritz values, sorted in nondecreasing order, which we compute using the Matlab command `eig`. We plot the convergence history of the relative distance of the largest or smallest Ritz value to the largest or smallest eigenvalue of A respectively, i.e., the quantities

$$\frac{|\lambda_{\max}(A) - \theta_k^{(k)}|}{\lambda_{\max}(A)}, \quad \frac{|\lambda_{\min}(A) - \theta_1^{(k)}|}{\lambda_{\min}(A)},$$

as a dash-dotted curve. The dashed and dotted curves are related to the relative accuracy of the estimates of the largest or smallest Ritz value,

$$\frac{|\theta_k^{(k)} - \text{est}_k^{\max}|}{\theta_k^{(k)}}, \quad \frac{|\theta_1^{(k)} - \text{est}_1^{\min}|}{\theta_1^{(k)}},$$

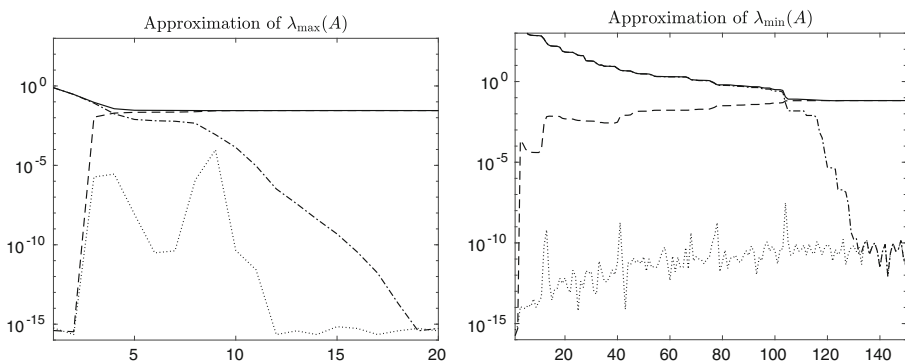


Fig. 3 Approximating the extreme Ritz values for the system bcsstk01

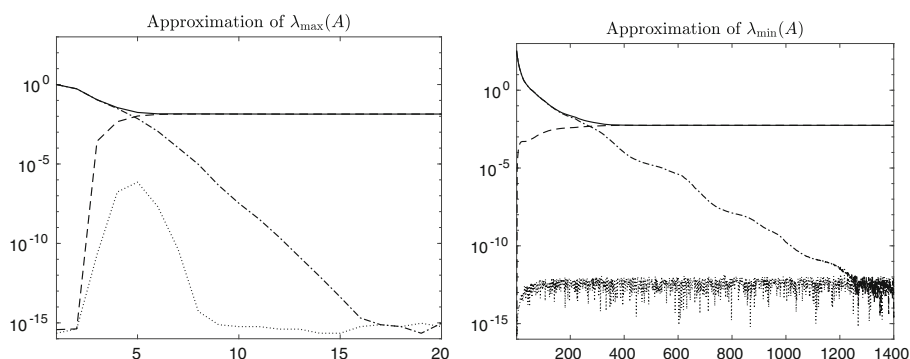


Fig. 4 Approximating the extreme Ritz values for the system Pb26

where est_k^{\max} stands for ρ_k^{\max} or $\hat{\rho}_k^{\max}$, and est_k^{\min} stands for ρ_k^{\min} or $\hat{\rho}_k^{\min}$. In particular, the dashed curves correspond to the relative accuracy of the cheap estimates ρ_k^{\max} and ρ_k^{\min} computed by Algorithms 4 and 5 respectively, while the dotted curve corresponds to the relative accuracy of the improved estimates $\hat{\rho}_k^{\max}$ and $\hat{\rho}_k^{\min}$, described in Sections 5.3 and 5.4. Finally, the relative distances of the cheap estimates ρ_k^{\max} and ρ_k^{\min} to the largest and smallest eigenvalues, i.e.,

$$\frac{|\lambda_{\max}(A) - \rho_k^{\max}|}{\lambda_{\max}(A)}, \quad \frac{|\lambda_{\min}(A) - \rho_k^{\min}|}{\lambda_{\min}(A)},$$

are plotted as a solid curve. Note that $\lambda_{\min}(A) < \theta_1^{(k)} \leq \text{est}_k^{\min}$ and $\text{est}_k^{\max} \leq \theta_k^{(k)} < \lambda_{\max}(A)$.

In Figs. 3 and 4, we can observe that if CG is applied to an *unpreconditioned* system, the largest Ritz values $\theta_k^{(k)}$ converge to $\lambda_{\max}(A)$ after a few iterations of CG (dash-dotted curve in the left part), while convergence of the smallest Ritz values $\theta_1^{(k)}$ to $\lambda_{\min}(A)$ (dash-dotted curve in the right part) is often delayed, and it is usually related to the convergence of the A -norm of the error.

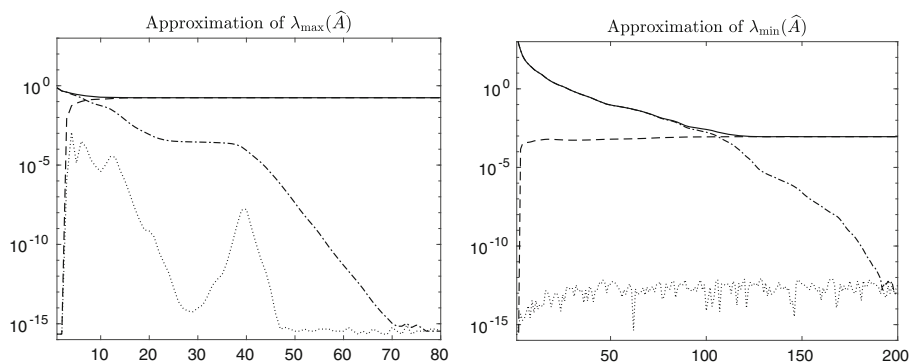


Fig. 5 Approximating the extreme Ritz values for the preconditioned system Pres_Poisson

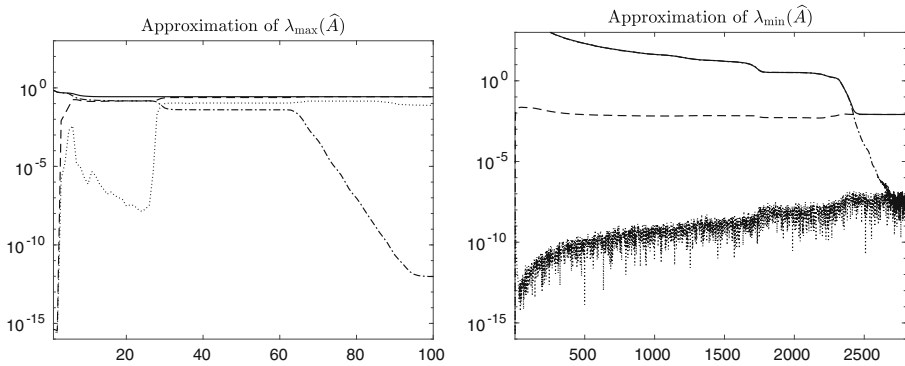


Fig. 6 Approximating the extreme Ritz values for the preconditioned system $s3dk t3m2$

In a few initial iterations, the cheap estimates ρ_k^{\max} and ρ_k^{\min} (dashed curves) approximate the corresponding Ritz values with a very high accuracy (in theory, the estimates agree with the exact Ritz values in iterations 1 and 2). However, in later iterations, their relative accuracy stagnates on the level of 10^{-1} or 10^{-2} . In other words, the estimates agree with the corresponding Ritz values to 1 or 2 valid digits. Since the extreme Ritz values approximate the extreme eigenvalues of A , the estimates also approximate these eigenvalues. We can observe that if an extreme Ritz value has converged, then its cheap estimate approximates the corresponding extreme eigenvalue to 1 or 2 valid digits (solid curve). Note that in most applications, this would be a sufficient accuracy. The dotted curves show the relative accuracy of the improved estimates $\hat{\rho}_k^{\max}$ and $\hat{\rho}_k^{\min}$ of the corresponding extreme Ritz values. The experiments predict that at the cost of computing one linear system with the tridiagonal matrix available in the form of LDL^T factorization, the accuracy of the estimates can be significantly improved.

A similar picture can be seen for preconditioned systems (see Figs. 5 and 6). Recall that if we precondition the system, the extreme Ritz values approximate the extreme eigenvalues of the *preconditioned* matrix \hat{A} . As we can see, convergence of $\theta_k^{(k)}$ to $\lambda_{\max}(\hat{A})$ to full precision accuracy is for the preconditioned systems significantly delayed. This is due to the fact that the preconditioned matrix has often a cluster of eigenvalues, which corresponds to the largest eigenvalue. Then, the power method as well as the Lanczos method (or CG) need more iterations to approximate the largest eigenvalue accurately. Moreover, a cluster of eigenvalues about the largest eigenvalue leads to a cluster of Ritz values which approximate the largest eigenvalue, and, as a consequence, the improved estimates $\hat{\rho}_k^{\max}$ (dotted curve) based on inverse iterations often do not improve the accuracy of the approximation significantly.

Similarly as in the unpreconditioned case, the cheap estimates ρ_k^{\max} and ρ_k^{\min} approximate the corresponding Ritz values with a high relative accuracy in a few initial iterations (dashed curve), but in later iterations, the relative accuracy is getting worse and stagnates on the level of about 10^{-1} or 10^{-2} . As a result, one can expect that in later iterations, the estimates ρ_k^{\max} and ρ_k^{\min} can approximate the largest

and smallest eigenvalues of \widehat{A} also with the relative accuracy of 10^{-1} or 10^{-2} (solid curve).

Note that in all our numerical experiments with various matrices, we have always observed that the estimates ρ_k^{\max} and ρ_k^{\min} approximate the largest and smallest eigenvalues to at least 1 or 2 digits of accuracy.

Finally, let us test numerically, how well the quantity $\xi_k^{1/2}$ approximates $\|x_k\|$ in the unpreconditioned case, and $\|x_k\|_M$ in the preconditioned case. Recall that ξ_k is defined by (33) and, in the experiments, we compute it cheaply using the formulas (34)–(35).

In Fig. 7, we consider the unpreconditioned systems `bcsstk01` and `Pb26`. By the dashed curve, we plot the relative error of the approximation

$$\left| \frac{\|x_k\| - \xi_k}{\|x_k\|} \right| \quad (\text{dashed}).$$

In the left part (system `bcsstk01`), the abovementioned relative error is close or below the level of 10^{-10} , despite the severe loss of orthogonality. In other words, ξ_k agrees with the approximated quantity $\|x_k\|$ to about 10 valid digits. For comparison, we also plot with a solid curve the relative error of $\|x_k\|$ as an approximation of $\|x\|$, and by dots the relative error of ξ_k as an approximation of $\|x\|$,

$$\left| \frac{\|x\| - \|x_k\|}{\|x\|} \right| \quad (\text{solid}), \quad \left| \frac{\|x\| - \xi_k}{\|x\|} \right| \quad (\text{dots}).$$

We can observe that the solid curve coincides visually with the dots until the level of 10^{-10} is reached. Below this level, the two curves can differ, but they are still close to each other. In the right part of the figure (system `Pb26`), the relative accuracy of ξ_k as an approximation of $\|x_k\|$ is even better, close to machine precision.

In Fig. 8, we consider systems `Pres_Poisson` and `s3dkt3m2` solved with preconditioning. Here, ξ_k is computed using the formulas (34)–(35) from the PCG coefficients $\widehat{\gamma}_k$ and $\widehat{\delta}_k$, and it approximates $\|x_k\|_M$. Similarly as for the

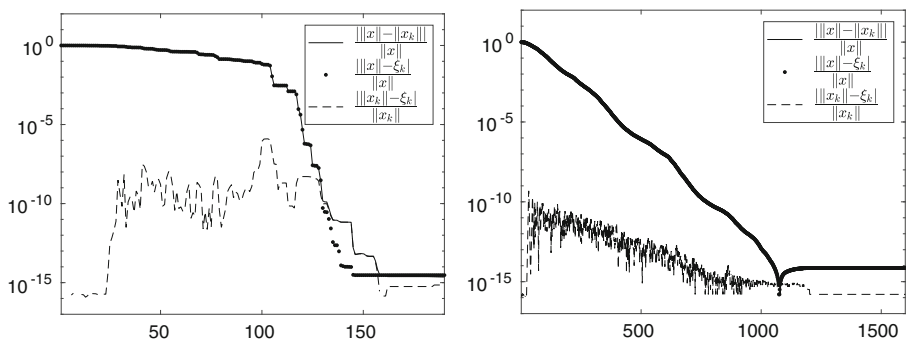


Fig. 7 Approximating $\|x_k\|$ using ξ_k when solving the unpreconditioned systems `bcsstk01` (left part) and `Pb26` (right part)

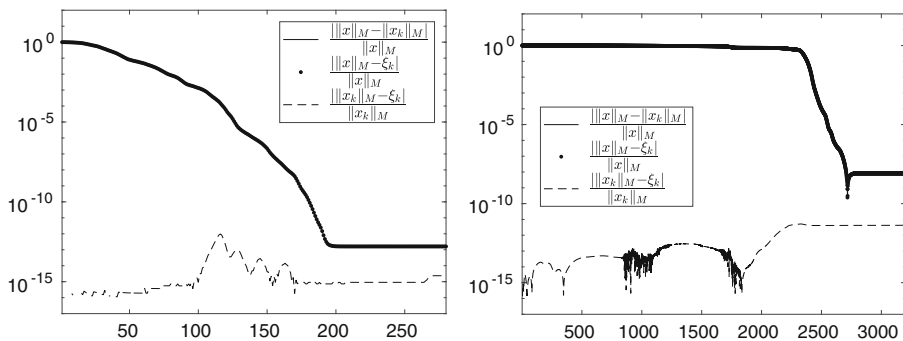


Fig. 8 Approximating $\|x_k\|_M$ using ξ_k when solving the preconditioned systems Pres_Poisson (left part) and s3dkt3m2 (right part)

unpreconditioned systems, we can observe that ξ_k approximates $\|x_k\|_M$ very accurately. In the considered examples, the relative errors are close to the level of machine precision.

8.2 Approximating convergence characteristics

The cheap approximations to the smallest and largest Ritz values, and to the norms of approximate solutions can be used to approximate various characteristics which provide some information about the convergence. In particular, in this section, we concentrate on approximating the normwise backward error and the Gauss-Radau upper bound, for the preconditioned systems Pres_Poisson and s3dkt3m2.

In Section 7, we discussed approximation of the normwise backward error. In Fig. 9, we plot the backward error (31) (solid curve) which corresponds to the original system, and the backward error (37) (dash-dotted curve) which corresponds to the preconditioned system. As mentioned in Section 7.2, using the cheap techniques, we can approximate the norm of the preconditioned matrix \tilde{A} , and the M -norm of

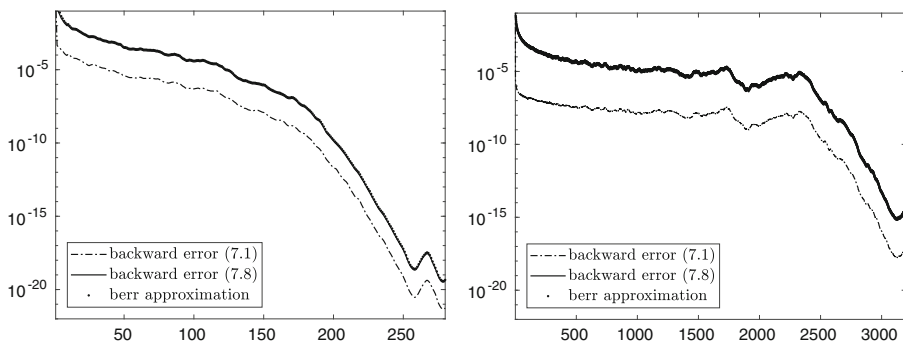


Fig. 9 Approximating the backward error for the preconditioned systems Pres_Poisson (left part) and s3dkt3m2 (right part)

the approximate solution $\|x_k\|_M$. Therefore, we can only efficiently approximate the backward error (37). The dots in Fig. 9 correspond to the approximations of the backward error (37), where $\|\hat{A}\|$ was approximated using the incremental technique (Algorithm 4) and $\|x_k\|_M$ was computed using the formulas (34)–(35). For both systems, we can observe that the backward error (37) visually coincides with its approximation.

The Gauss-Radau upper bound can be approximated using the approximation (30) which does not require any a priori information about the smallest eigenvalue (see Section 6). In Fig. 10, we plot the A -norm of the error (dotted curve) and the Gauss-Radau upper bounds $u_k^{(\mu)}$ (dashed curves), where the values of μ closely approximate the smallest eigenvalue of the preconditioned matrix \hat{A} from below. Similarly as in Section 4, we choose μ to be equal to

$$\frac{\lambda_{\min}(\hat{A})}{(1 + 10^{-m})}, \quad \text{for } m = 1, 4, 8, 12.$$

The approximation (30) using μ_k is plotted as a solid curve. As expected, the quantity (30) underestimates the A -norm of the error in the initial stage of convergence, since the smallest Ritz value is a poor approximation to the smallest eigenvalue. However, as soon as the smallest Ritz value approximates the smallest eigenvalue, the quantity (30) bounds the A -norm of the error from above. Moreover, in the final stage of convergence, the quantity (30) is as good as the Gauss-Radau upper bounds even if μ approximates $\lambda_{\min}(\hat{A})$ tightly. As in the numerical example presented in Section 4, we can observe that the Gauss-Radau upper bounds are very sensitive to the accuracy to which μ approximates $\lambda_{\min}(\hat{A})$. Nevertheless, below some level, all the values of μ give visually the same upper bound, which is not very close to the A -norm of the error. This phenomenon appeared almost in all experiments we performed and we believe it deserves further investigation.

In the last numerical experiment (Fig. 11), we choose the delay $d = 10$ for the preconditioned system `Pres_Poisson` and $d = 40$ for `s3dkt3m2`. We approximate the A -norm of the error (dotted curve) using the Gauss-Radau upper bound

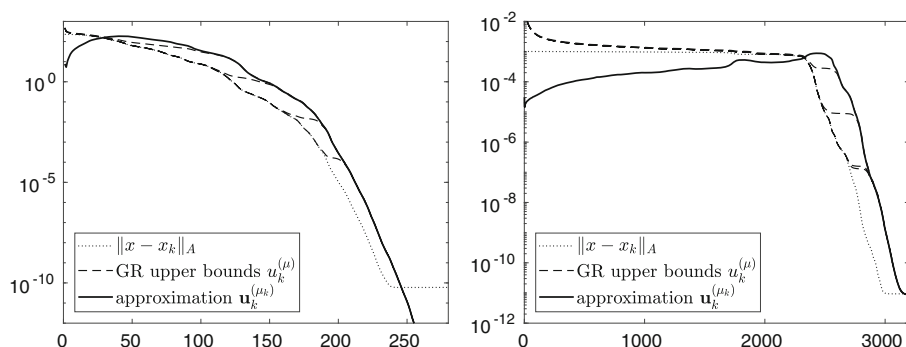


Fig. 10 Approximating the Gauss-Radau upper bound for the preconditioned systems `Pres_Poisson` (left part) and `s3dkt3m2` (right part)

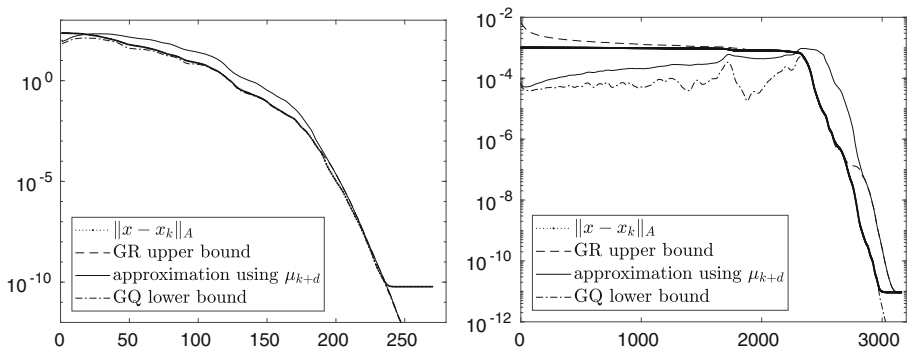


Fig. 11 Approximating the A -norm of the error using Gauss-Radau upper bound, the approximation (30), and the Gauss lower bound for the preconditioned systems `Pres_Poisson` (left part, $d = 10$) and `s3dkt3m2` (right part, $d = 40$)

(10) (dashed curve) for a value of μ which closely approximates $\lambda_{\min}(\hat{A})$ from below, $\mu = \lambda_{\min}(\hat{A})/(1 + 10^{-12})$, simulating the situation when we know $\lambda_{\min}(\hat{A})$ in advance from the application. If there is no a priori information about $\lambda_{\min}(\hat{A})$, one can use the approximation of the upper bound (15) (solid curve) with μ_{k+d} . For comparison, we also plot the Gauss lower bound based on (9) (dash-dotted curve).

In the left part of Fig. 11, we can observe that $d = 10$ significantly improves all the bounds. The approximation of the upper bound (15) (solid) is slightly overestimating the A -norm of the error $\|x - x_k\|_A$ in the initial stage of convergence. When convergence accelerates (around iteration 200), all the estimates approximate $\|x - x_k\|_A$ tightly. In Fig. 10 (left part), we have observed that the curves describing upper bounds are about 10 iterations delayed in the later stage of convergence. This is the reason why the choice of $d = 10$ is sufficient to get good approximations to $\|x - x_k\|_A$.

In the right part of Fig. 11, we consider the more complicated problem with the system `s3dkt3m2`. Here, the choice of $d = 40$ does not improve the bounds too much in the initial stagnation phase. The Gauss lower bound (dash-dotted) as well as the approximation of the upper bound (15) (solid) underestimate $\|x - x_k\|_A$ significantly. The only useful bound in this phase of convergence is the Gauss-Radau upper bound (dashed) with a prescribed value of μ . When the A -norm of the error starts to decrease (around iteration 2300), the Gauss lower bound with $d = 40$ starts to be visually the same as $\|x - x_k\|_A$, until the ultimate level of accuracy is reached. This is not the case for the approximation of the upper bound (15) (solid), which is significantly delayed. However, in comparison to Fig. 10 (right part), the approximation of (15) is moved about 40 iterations towards $\|x - x_k\|_A$. The Gauss-Radau upper bound (dashed) approximates at first $\|x - x_k\|_A$ tightly, but, below a certain level, it starts to give the same results as the approximation of (15), i.e., the curve is delayed. This experiment demonstrates the potential weakness of upper bounds and the approximation of (15) in the final stage of convergence. It also shows a need for an adaptive choice of d at each iteration, which would reflect the desired accuracy of approximations.

9 Conclusions

In this paper, we derived a new upper bound for the A -norm of the error in CG. The new bound is closely related to the Gauss-Radau upper bound. While the Gauss-Radau upper bound can be very sensitive to the choice of the parameter μ which should closely approximate the smallest eigenvalue of the (preconditioned) system matrix from below, the new bound is not sensitive to the choice of μ . One can use it even if μ is larger than the smallest eigenvalue, as an approximation of the upper bound, so that μ can be chosen as an approximation to the smallest Ritz value.

We next developed a very cheap algorithm for approximating the smallest and largest Ritz values during the CG computations. These approximations can further be improved using inverse iterations, at the cost of storing the CG coefficients and solving a linear system with a tridiagonal matrix at each CG iteration. The cheap approximations to the smallest and largest Ritz values can be useful in general, e.g., to approximate almost for free the condition number of the system matrix, or to estimate the ultimate level of accuracy. In this paper, we used them to approximate the parameter μ for the new upper bound on the A -norm of the error, and also to approximate the 2-norm of the system matrix when computing the normwise backward error.

Numerical experiments show that the approximation of the upper bound for the A -norm of the error which uses the cheap technique to approximate the smallest Ritz value is in the later stage of convergence usually as good as the Gauss-Radau upper bound for which μ has to be prescribed. We also observed that even if the smallest eigenvalue is known in advance, the Gauss-Radau upper bound loses its sharpness as the A -norm of the error decreases, and, below some level, it is visually the same as its approximation (30). This phenomenon is caused by the underlying finite precision Lanczos process, and it deserves additional investigation.

As further demonstrated, the quality of the lower and upper bounds can be improved using the delay parameter d . This technique is very promising for practical estimation of the A -norm of the error in CG. However, constant value of d is usually not sufficient in the initial stage of convergence, and it requires too many extra steps of CG in the convergence phase. Hence, there is a need for developing a heuristic technique to choose d adaptively at each iteration, to reflect the required accuracy of the estimate. We believe that results of this paper can be useful in developing such a technique. The adaptive choice of d remains a subject of our further work.


Acknowledgments The authors thank an anonymous referee for the very helpful comments.

References

1. Arioli, M., Duff, I.S., Ruiz, D.: Stopping criteria for iterative solvers. *SIAM J. Matrix Anal. Appl.* **13**(1), 138–144 (1992)
2. Barrett, R., Berry, M., Chan, T.F., et al.: *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (1994)
3. Bischof, C.H.: Incremental condition estimation. *SIAM J. Matrix Anal. Appl.* **11**(2), 312–322 (1990)

4. Dahlquist, G., Eisenstat, S.C., Golub, G.H.: Bounds for the error of linear systems of equations using the theory of moments. *J. Math. Anal. Appl.* **37**, 151–166 (1972)
5. Dahlquist, G., Golub, G.H., Nash, S.G.: Bounds for the error in linear systems. In: *Semi-Infinite Programming* (Proc. Workshop, Bad Honnef, 1978), *Lecture Notes in Control and Information Sci.*, vol. 15, pp. 154–172. Springer, Berlin (1979)
6. Duff, I.S., Vömel, C.: Incremental norm estimation for dense and sparse matrices. *BIT* **42**(2), 300–322 (2002)
7. Duintjer Tebbens, J., Tüma, M.: On incremental condition estimators in the 2-norm. *SIAM J. Matrix Anal. Appl.* **35**(1), 174–197 (2014)
8. Eiermann, M., Ernst, O.G.: Geometric aspects of the theory of Krylov subspace methods. *Acta Numer.* **10**, 251–312 (2001)
9. Fischer, B., Golub, G.H.: On the error computation for polynomial based iteration methods. In: *Recent Advances in Iterative Methods*, IMA Vol. Math. Appl., Vol. 60, pp. 59–67. Springer, New York (1994)
10. Golub, G.H., Meurant, G.: Matrices, moments and quadrature. In: *Numerical Analysis 1993* (Dundee, 1993), *Pitman Res. Notes Math. Ser.*, Vol. 303, pp. 105–156. Longman Sci. Tech., Harlow (1994)
11. Golub, G.H., Meurant, G.: Matrices, moments and quadrature. II. How to compute the norm of the error in iterative methods. *BIT* **37**(3), 687–705 (1997)
12. Golub, G.H., Meurant, G.: *Matrices, Moments and Quadrature with Applications*. Princeton University Press, Princeton (2010)
13. Golub, G.H., Strakoš, Z.: Estimates in quadratic formulas. *Numer. Algorithms* **8**(2–4), 241–268 (1994)
14. Golub, G.H., Van Loan, C.F. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences, 4th edn. Johns Hopkins University Press, Baltimore (2013)
15. Greenbaum, A.: Estimating the attainable accuracy of recursively computed residual methods. *SIAM J. Matrix Anal. Appl.* **18**(3), 535–551 (1997)
16. Gutknecht, M.H., Rozložník, M.: By how much can residual minimization accelerate the convergence of orthogonal residual methods? *Numer. Algorithms* **27**(2), 189–213 (2001)
17. Gutknecht, M.H., Rozložník, M.: Residual smoothing techniques: do they improve the limiting accuracy of iterative solvers? *BIT* **41**(1), 86–114 (2001)
18. Hestenes, M.R., Stiefel, E.: Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand.* **49**, 409–436 (1952)
19. Higham, N.J.: *Accuracy and stability of numerical algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (1996)
20. Kouhia, R.: Description of the CYLSHELL set. Laboratory of Structural Mechanics, Finland (1998)
21. Meurant, G.: The computation of bounds for the norm of the error in the conjugate gradient algorithm. *Numer. Algorithms* **16**(1), 77–87 (1998)
22. Meurant, G.: *The Lanczos and Conjugate Gradient Algorithms, from Theory to Finite Precision Computations, Software, Environments and Tools*, vol. 19. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2006)
23. Meurant, G., Tichý, P.: On computing quadrature-based bounds for the A -norm of the error in conjugate gradients. *Numer. Algo.* **62**(2), 163–191 (2013)
24. Meurant, G., Tichý, P.: Erratum to: On computing quadrature-based bounds for the A -norm of the error in conjugate gradients [mr3011386]. *Numer. Algorithms* **66**(3), 679–680 (2014)
25. Oettli, W., Prager, W.: Compatibility of approximate solution of linear equations with given error bounds for coefficients and right-hand sides. *Numer. Math.* **6**(1), 405–409 (1964)
26. Paige, C.C., Saunders, M.A.: LSQR: an algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Software* **8**(1), 43–71 (1982)
27. Parlett, B.N., Dhillon, I.S.: Relatively robust representations of symmetric tridiagonals. In: *Proceedings of the International Workshop on Accurate Solution of Eigenvalue Problems* (University Park, PA, 1998), vol. 309, pp. 121–151 (2000)
28. Rigal, J.L., Gaches, J.: On the compatibility of a given solution with the data of a linear system. *Journal of the ACM (JACM)* **14**(3), 543–548 (1967)
29. Strakoš, Z., Tichý, P.: On error estimation in the conjugate gradient method and why it works in finite precision computations. *Electron. Trans. Numer. Anal.* **13**, 56–80 (2002)
30. Strakoš, Z., Tichý, P.: Error estimation in preconditioned conjugate gradients. *BIT* **45**(4), 789–817 (2005)

Affiliations

G rard Meurant¹ · Petr Tich  ² 

G rard Meurant

¹ Paris, France

² Faculty of Mathematics and Physics, Charles University, Sokolovsk   83, 186 75 Prague 8, Czech Republic