Proceedings of ALGORITMY 2016
pp. 323–332

ON ERROR ESTIMATION IN THE CONJUGATE GRADIENT METHOD: NORMWISE BACKWARD ERROR[*]

PETR TICHÝ[†]

Abstract. Using an idea of Duff and Vömel [BIT, 42 (2002), pp. 300–322] we suggest a simple algorithm that incrementally estimates the 2-norm of Jacobi matrices that are available during the conjugate gradient (CG) computations. The estimate can be used, e.g., in stopping criteria based on the normwise backward error. Numerical experiments predict that the estimate approximates the 2-norm of A with a sufficient accuracy.

Key words. conjugate gradients, normwise backward error, incremental norm estimation.

AMS subject classifications. 65F10, 65F35.

1. Introduction.

The (preconditioned) Conjugate Gradient (CG) algorithm by Hestenes and Stiefel [8] is the iterative method of choice for solving linear systems $Ax = b$ with a real symmetric positive definite matrix A. An important question is when to stop the iterations.

The choice of the stopping criterion typically depends on the underlying problem. Ideally, one would like to stop the iterations when some norm of the error $x-x_k$, where x_k are the CG iterates, is small enough. The norm of the error which is particularly interesting for CG is the A-norm which is minimized at each iteration,

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

Hestenes and Stiefel [8] considered the A-norm of the error a possible candidate for measuring the "goodness" of x_k as an estimate of x. They showed that though it was impossible to compute the A-norm of the kth error without knowing the solution x, it was possible to estimate it. Later, inspired by the connection of CG with the Gauss quadrature rule for a Riemann-Stieltjes integral, a way of research on this topic was started by Gene Golub in the 1970s and continued throughout the years with several collaborators (e.g., Dahlquist, Eisenstat, Fischer, Meurant, Strakoš); see, e.g., [6]. Note that quadrature rules can also be used for estimating the Euclidean norm of the error $\|x - x_k\|$ that plays an important role in the applications such as in image processing; see, e.g., [11, 6].

Denote the matrix 2-norm by $\|A\|$. In [12, 1], backward error perturbation theory was used to derive a family of stopping criteria for iterative methods. In particular, given x_k, one can ask what are the norms of the smallest perturbations ΔA of A and Δb of b measured in the relative sense such that the approximate solution x_k represents the exact solution of the perturbed system

$$(A + \Delta A)\, x_k = b + \Delta b\,.$$

In other words, we are interested in the quantity

$$\min\left\{\varepsilon :\; (A + \Delta A)\, x_k = b + \Delta b,\; \|\Delta A\| \le \varepsilon\|A\|,\; \|\Delta b\| \le \varepsilon\|b\|\right\}\,.$$

[*] This work was supported by the GACR grant P201/13-06684 S.

[†] Faculty of Mathematics and Physics, Charles University in Prague & Institute of Computer Science, Czech Academy of Sciences (tichy@cs.cas.cz).

It was show by Rigal and Gaches [13] that this quantity, called *the normwise backward error*, is given by

$$\frac{\|r_k\|}{\|A\|\|x_k\| + \|b\|}\,. \tag{1.1}$$

where $r_k = b - Ax_k$. This approach can be generalized in order to quantify levels of confidence in A and b, see [12, 1]. The normwise backward error is, as a base for stopping criteria, frequently recommended in the numerical analysis literature, see, e.g. [9, 2].

In this work we concentrate on the question how to efficiently estimate the quantity (1.1) during the CG computations. While $\|b\|$, $\|r_k\|$, and $\|x_k\|$ are available, $\|A\|$ has to be computed or estimated, which could represent an extra work. To approximate $\|A\|$, one can use, for example, matrix norms inequalities, in particular,

$$\frac{1}{\sqrt{n}}\,\|A\|_F \le \|A\| \le \|A\|_F$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix. Having lower and upper bounds on $\|A\|$, we obtain lower and upper bounds on the quantity (1.1). However, for large n, these bounds need not approximate (1.1) well, and it is desirable to get a better approximation of $\|A\|$.

Since A is symmetric and positive definite, $\|A\|$ is equal to the maximum eigenvalue of A. Hence, a natural way to approximate $\|A\|$ during the CG computations is to use maximum Ritz values that can be determined from the CG coefficients. Note that the maximum Ritz value is the maximum eigenvalue of the corresponding Jacobi matrix. Computing the eigenvalues of Jacobi matrices at each iteration would be too expensive. Instead of it, one can compute the maximum Ritz value only once at some iteration k, and use it as an approximation of $\|A\|$. This approach requires to determine the iteration k by monitoring the convergence of the maximum Ritz value to the maximum eigenvalue of A, and to compute the maximum eigenvalue of the tridiagonal matrix. All that can be done in $\mathcal{O}(k^2)$ operations and it works well, in general. However, the mentioned approach represents an algorithmic complication for a potential user. In this paper we present a very simple way to approximate the maximum Ritz value incrementally at a negligible cost. The user needs to add just a few lines into his code.

The outline of the paper is as follows. Section 2 recalls some basic facts about the Lanczos and CG algorithms. In Section 3 we show how to incrementally estimate the maximum Ritz value, and formulate the algorithm. Finally, in Section 4 we present numerical experiments.

2. CG and Lanczos algorithms.

In this section we briefly recall the Lanczos and Conjugate Gradient algorithms as well as their relationships; see, for instance [7, 10].

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

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

called Krylov subspaces. The dimension of these subspaces is increasing up to an index n called the grade of v with respect to A, at which the maximum dimension is attained, and $\mathcal{K}_n(A, v)$ is invariant under multiplication with A. Assuming that $k < n$

Algorithm 1 Lanczos algorithm

```
input $A$, $v$
$\beta_0 = 0$, $v_0 = 0$
$v_1 = v/\|v\|$
for $k = 1, \ldots$ do
    $w = Av_k - \beta_{k-1}v_{k-1}$
    $\alpha_k = v_k^T w$
    $w = w - \alpha_k v_k$
    $\beta_k = \|w\|$
    $v_{k+1} = w/\beta_k$
end for
```

the Lanczos algorithm (Algorithm 1) computes an orthonormal basis $v_1, \ldots, v_{k+1}$ of the Krylov subspace $\mathcal{K}_{k+1}(A, v)$. The basis vectors v_j satisfy the matrix relation

$$AV_k = V_k T_k + \beta_{k+1} v_{k+1} e_k^T$$

where $V_k = [v_1 \cdots v_k]$ and T_k is the $k \times k$ symmetric tridiagonal matrix of the recurrence coefficients computed in Algorithm 1:

$$T_k = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \ddots & & \\ & & \ddots & \beta_{k-1} \\ & & \beta_{k-1} & \alpha_k \end{bmatrix}.$$

The coefficients β_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.

Algorithm 2 Conjugate gradients

```
input $A$, $b$, $x_0$
$r_0 = b - Ax_0$
$p_0 = r_0$
for $k = 1, \ldots, n$ 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
```

When solving a system of linear algebraic equations $Ax = b$ with symmetric and positive definite matrix A, the CG method (Algorithm 2) can be used. CG computes iterates x_k that are optimal since the A-norm of the error is minimized over the manifold $x_0 + \mathcal{K}_k(A, r_0)$,

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

The residual vectors r_k are proportional to the Lanczos basis vectors v_j and hence mutually orthogonal,

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

Therefore, the residual vectors r_j yield an orthogonal basis of the Krylov subspace $\mathcal{K}_{k+1}(A, r_0)$. In this sense, CG can be seen as an algorithm for computing an orthogonal basis of the Krylov subspace $\mathcal{K}_{k+1}(A, r_0)$ and there is a close relationship between the CG and Lanczos algorithms. It is well-known (see, for instance [10]) that the recurrence coefficients computed in both algorithms are connected via

$$\beta_k = \frac{\sqrt{\delta_k}}{\gamma_{k-1}}, \quad \alpha_k = \frac{1}{\gamma_{k-1}} + \frac{\delta_{k-1}}{\gamma_{k-2}}, \quad \delta_0 = 0, \quad \gamma_{-1} = 1,$$

writing these formulas in a matrix form, we get

$$T_k = R_k^T R_k$$

where

$$R_k = \left[\begin{array}{cccc} \frac{1}{\sqrt{\gamma_0}} & \sqrt{\frac{\delta_1}{\gamma_0}} & & \\ & \ddots & \ddots & \\ & & \ddots & \sqrt{\frac{\delta_{k-1}}{\gamma_{k-2}}} \\ & & & \frac{1}{\sqrt{\gamma_{k-1}}} \end{array} \right].$$

In other words, CG computes implicitly the Cholesky factorization of the Jacobi matrix T_k generated by the Lanczos algorithm. Since for any $y \in \mathbb{R}^k$ we have $y^T T_k y = \|R_k y\|^2$, it holds that

$$\|T_k\| = \|R_k\|^2.$$

We can see that to approximate the maximum eigenvalue of T_k, one can use algorithms that incrementally approximate the maximum singular value of the upper triangular matrix R_k.

3. The incremental approximation of $\|T_k\|$.

As mentioned above, $\|A\|$ can be approximated using $\|T_k\| = \|R_k\|^2$. To approximate the maximum singular value of R_k, we use the incremental estimator for tridiagonal matrices proposed in [4], and specialize the algorithm for the case when R_k is only bidiagonal. The algorithm in [4] is based on incremental improvement of an approximation of the right singular vector that corresponds to the maximum singular value. In [5] 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]. As we will see, in our case, we need to store just the last component of the approximate maximum right singular vector.

3.1. 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. Let

$$B = \left[\begin{array}{cc} \alpha & \beta \\ \beta & \gamma \end{array} \right], \tag{3.1}$$

then the eigenvalues of B are given by, see, e.g., [4, p. 306],

$$\lambda_+ = \frac{1}{2}(\alpha + \gamma + \omega), \qquad \lambda_- = \frac{1}{2}(\alpha + \gamma - \omega)$$

where

$$\omega^2 = (\alpha - \gamma)^2 + 4\beta^2.$$

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

$$\begin{bmatrix} \alpha - \gamma + \omega & \alpha - \gamma - \omega \\ 2\beta & 2\beta \end{bmatrix}.$$

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

$$\hat{R} = \begin{bmatrix} R & v \\ & \mu \end{bmatrix}, \qquad v \in \mathbb{R}^k, \quad \mu \in \mathbb{R}, \tag{3.2}$$

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

$$\hat{z} = \begin{bmatrix} sz \\ c \end{bmatrix}, \tag{3.3}$$

where $s^2 + c^2 = 1$ are chosen such that the norm of the vector

$$\hat{R}\hat{z} = \begin{bmatrix} sRz + cv \\ c\mu \end{bmatrix}$$

is maximum. It holds that

$$\|\hat{R}\hat{z}\|^2 = s^2\alpha + 2sc\beta + c^2\gamma = \begin{bmatrix} s \\ c \end{bmatrix}^T \begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix} \begin{bmatrix} s \\ c \end{bmatrix}$$

where

$$\alpha = \|Rz\|^2, \qquad \beta = v^T Rz, \qquad \gamma = v^T v + \mu^2. \tag{3.4}$$

Hence, to maximize $\|\hat{R}\hat{z}\|^2$, we need to determine the maximum eigenvalue of the 2×2 matrix (3.1), and the corresponding eigenvector. Using the previous results,

$$\begin{bmatrix} s \\ c \end{bmatrix} = \frac{u}{\|u\|}, \qquad u = \begin{bmatrix} \alpha - \gamma + \omega \\ 2\beta \end{bmatrix}$$

and

$$\lambda_{\max} = \frac{\alpha + \gamma + \omega}{2}, \qquad \omega^2 = (\alpha - \gamma)^2 + 4\beta^2.$$

Note that if $\beta = 0$, the formula for the eigenvector that corresponds to $\lambda_{\max}$ is still valid. Next, it holds that

$$\|u\|^2 = 2(\omega^2 + (\alpha - \gamma)\,\omega),$$

and, therefore,

$$c^2 = \frac{2\beta^2}{\omega^2 + (\alpha-\gamma)\,\omega} = \frac{1}{2}\frac{\omega^2 - (\alpha-\gamma)^2}{\omega^2 + (\alpha-\gamma)\,\omega} = \frac{1}{2}\left(1 - \frac{\alpha-\gamma}{\omega}\right).$$

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

$$\|\hat{R}\hat{z}\|^2 = \frac{\alpha+\gamma+\omega}{2} = \alpha + \frac{\omega}{2}\left(1 - \frac{\alpha-\gamma}{\omega}\right) = \alpha + \omega c^2.$$

In summary,

$$c^2 = \frac{1}{2}\left(1 - \frac{\alpha-\gamma}{\omega}\right), \qquad \|\hat{R}\hat{z}\|^2 = \alpha + \omega c^2\,. \tag{3.5}$$

3.3. Specialization to upper bidiagonal matrices . We now apply the technique from the previous subsection to the bidiagonal matrices R_k which are available in CG. The previous technique requires to store the vector z and to perform $\mathcal{O}(k)$ operations per update. We will show that the incremental estimate of $\|R_k\|$ can be computed without storing the vector z, and in $\mathcal{O}(1)$ operations.

Consider the iteration $k{+}1$ of the CG algorithm and suppose that the approximate maximum right singular vector z_k of R_k is known. The new matrix R_{k+1} arises from R_k by adding one column and one row. Comparing the form of R_{k+1} with (3.2) and taking $R = R_k$, we see immediately that

$$v = \sqrt{\frac{\delta_k}{\gamma_{k-1}}}e_k, \qquad \mu = \frac{1}{\gamma_k} \tag{3.6}$$

and from (3.4) and (3.6),

$$\alpha = \|R_k z_k\|^2, \qquad \gamma = \frac{\delta_k}{\gamma_{k-1}} + \frac{1}{\gamma_k} = \alpha_{k+1}, \qquad \beta = \frac{\sqrt{\delta_k}}{\gamma_{k-1}}c_k = \beta_k^2 c_k^2$$

where c_k is the last component of the vector z_k (see (3.3)), and α_{k+1} and β_k are the Lanczos coefficients. Denote

$$\Delta_k = \|R_k z_k\|^2.$$

We show how to efficiently compute Δ_{k+1} and c_{k+1}. With the above notation, and using (3.5) we get

$$c_{k+1}^2 = \frac{1}{2}\left(1 - \frac{\Delta_k - \alpha_{k+1}}{\omega_k}\right), \qquad \Delta_{k+1} = \Delta_k + \omega_k c_{k+1}^2,$$

where

$$\omega_k^2 = (\Delta_k - \alpha_{k+1})^2 + 4\beta_k^2 c_k^2.$$

In summary, we can easily compute the estimate Δ_{k+1} of $\|T_{k+1}\|$ from the coefficients

γ_k and δ_k generated by the CG algorithm, using the folowing recurrences

$$\begin{aligned}
\alpha_{k+1} &= \frac{\delta_k}{\gamma_{k-1}} + \frac{1}{\gamma_k} \\
\beta_k^2 &= \frac{\delta_k}{\gamma_{k-1}^2} \\
\omega_k &= \sqrt{(\Delta_k - \alpha_{k+1})^2 + 4\beta_k^2 c_k^2} \\
c_{\mathrm{k}+1}^2 &= \frac{1}{2}\left(1 - \frac{\Delta_k - \alpha_{k+1}}{\omega_k}\right) \\
\Delta_{k+1} &= \Delta_k + \omega_k c_{k+1}^2 .
\end{aligned}$$

We have not specified yet, how to start the algorithm for computing the estimate Δ_k. For $k = 2$, we are able to determine the maximum eigenvalue of T_2 and the corresponding eigenvector exactly. Using results of Subsection 3.1, it holds that

$$c_2^2 = \frac{1}{2}\left(1 - \frac{\alpha_1 - \alpha_2}{\omega_1}\right), \qquad \text{where} \qquad \omega_1^2 = (\alpha_1 - \alpha_2)^2 + 4\beta_1^2,$$

and

$$\Delta_2 = \|R_2\|^2 = \|T_2\| = \frac{\alpha_1 + \alpha_2 + \omega_1}{2} = \alpha_1 + \omega_1 c_2^2.$$

We can see that the formulas for $k = 2$ will be consistent with formulas for a general k, if we define

$$\Delta_1 = \alpha_1, \qquad \text{and} \qquad c_1 = 1.$$

Finally, we get Algorithm 3 that computes the estimates Δ_k of $\|T_k\|$ almost for free. This estimate can be used to approximate $\|A\|$ and the normwise backward error. In particular, since $\Delta_k \le \|T_k\| \le \|A\|$, we get an upper bound

$$\varrho_k \equiv \frac{\|r_k\|}{\|A\|\|x_k\| + \|b\|} \le \frac{\|r_k\|}{\Delta_k\|x_k\| + \|b\|} \equiv \eta_k\,. \tag{3.7}$$

4. Numerical experiments. In the following numerical experiments performed in Matlab 8.0 (R2012b) we test the quality of the upper bound

$$\eta_k \equiv \frac{\|r_k\|}{\Delta_k\|x_k\| + \|b\|} \tag{4.1}$$

on the normwise backward error. We solve two systems of linear algebraic equations $Ax = b$, the first one with the matrix `bcsstk03` of order $n = 112$, and the second one with the matrix `bcsstk16` of order $n = 4884$. Both matrices can be found in the Harwell-Boeing collection. The right-hand side b has been chosen such that b has equal components in the eigenvector basis, and such that $\|b\| = 1$. We choose $x_0 = 0$. Results plotted in Figure 4.1 and Figure 4.2 correspond to the matrix `bcsstk03` and `bcsstk16`, respectively.

In both figures we observe that the normwise backward error ϱ_k (1.1) and its estimate η_k (4.1) computed by Algorithm 3 visually coincide. We know from (3.7) that η_k represents an upper bound on ϱ_k. Hence, when stopping the algorithm if η_k

Algorithm 3 CG with incremental norm estimation

```
input $A$, $b$, $x_0$
$r_0 = b - Ax_0$, $p_0 = r_0$
$\delta_0 = 0$, $\gamma_{-1} = 0$
for $k = 1, \ldots,$ do
    $\gamma_{k-1} = \frac{r_{k-1}^T r_{k-1}}{p_{k-1}^T A p_{k-1}}$,
    $x_k = x_{k-1} + \gamma_{k-1} p_{k-1}$
    $r_k = r_{k-1} - \gamma_{k-1} A p_{k-1}$
    $\delta_k = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}$
    $p_k = r_k + \delta_k p_{k-1}$
    $\alpha_k = \frac{1}{\gamma_{k-1}} + \frac{\delta_{k-1}}{\gamma_{k-2}}$
    $\beta_k^2 = \frac{\delta_k}{\gamma_{k-1}^2}$
    if $k = 1$ then
        $c_1^2 = 1$
        $\Delta_1 = \alpha_1$
    else
        $\omega_{k-1} = \sqrt{\left(\Delta_{k-1} - \alpha_k\right)^2 + 4\beta_{k-1}^2 c_{k-1}^2}$
        $c_k^2 = \frac{1}{2}\left(1 - \frac{\Delta_{k-1} - \alpha_k}{\omega_{k-1}}\right)$
        $\Delta_k = \Delta_{k-1} + \omega_{k-1} c_k^2$
    end if
end for
```

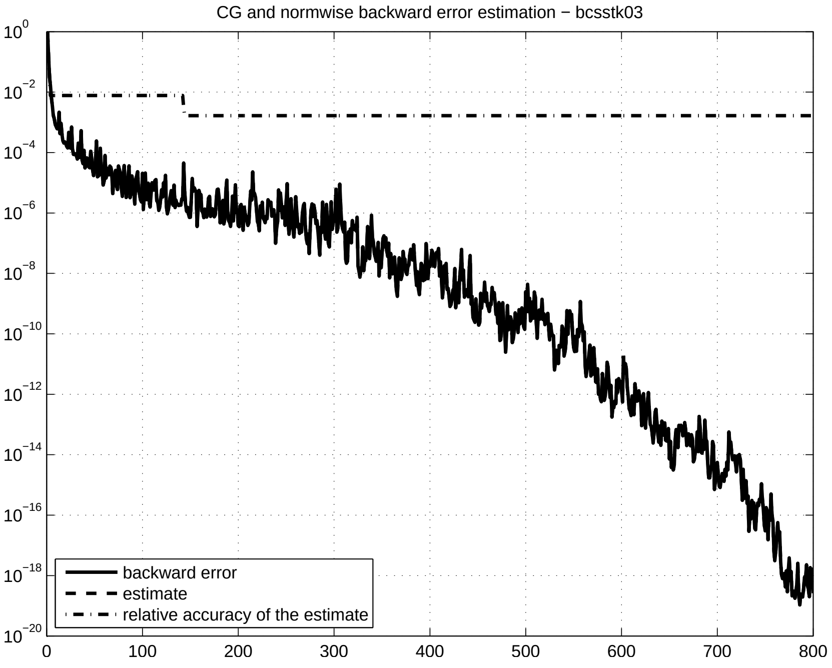

FIG. 4.1. *Normwise backward error in CG and its estimate for the matrix* bcsstk03 *of order* $n = 112$. *The normwise backward error (solid line) and its estimate (dashed line) visually coincide. By the dash-dotted line we plot the relative error of the estimate given by the quantity (4.2).*

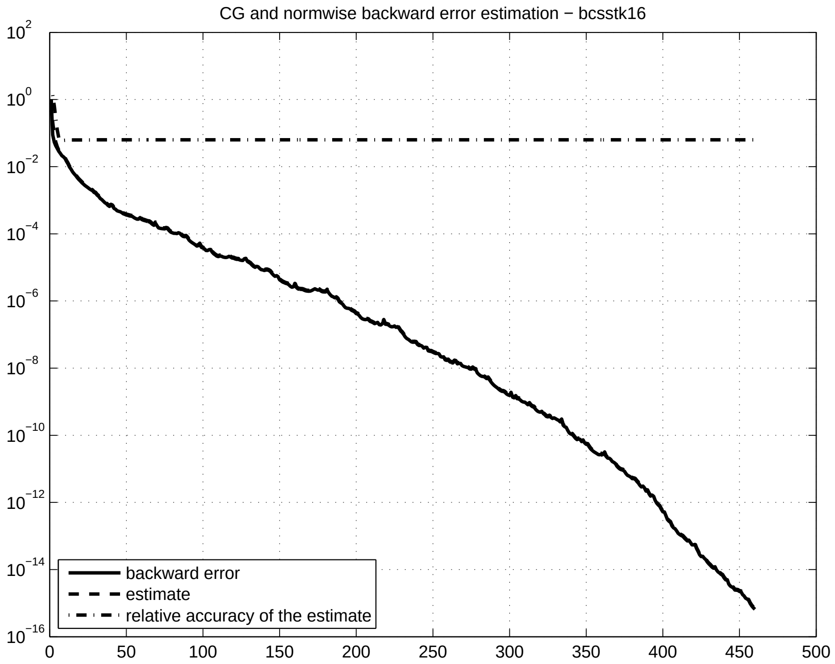

FIG. 4.2. *Normwise backward error in CG and its estimate for the matrix* `bcsstk16` *of order* $n = 4884$. *The normwise backward error (solid line) and its estimate (dashed line) visually coincide. By the dash-dotted line we plot the relative error of the estimate given by the quantity (4.2).*

is less than a given tolerance, then also ϱ_k is less than this tolerance. Moreover, since both quantities have the same order of magnitude, the CG algorithm will be stopped using η_k at the right iteration.

The two quantities η_k and ϱ_k are not only of the same magnitude, but they even agree to 2 or 3 valid digits, which is demonstrated by plotting the ratio

$$\frac{\eta_k - \varrho_k}{\varrho_k} \tag{4.2}$$

that provides information about the relative accuracy of the estimate (dash-dotted line). When increasing the number of iterations, the relative accuracy of the estimate η_k stagnates. This can be explained by the fact that the estimator of the matrix 2-norm has been suggested to be computationally cheap and efficient, but, on the other hand, one cannot expect a very high relative accuracy of the matrix norm estimate in general. If one needs from some reason to improve the accuracy of the estimate η_k, one can compute the maximum eigenvalue of T_k and the corresponding eigenvector for some larger value of k, and then start the incremental estimation of the matrix 2-norm using the formulas presented in Subsection 3.3.

Acknowledgments. The author thanks Jurjen Duintjer Tebbens and the anonymous referee for very helpful comments.

REFERENCES

[1] M. Arioli, I. Duff, and D. Ruiz. Stopping criteria for iterative solvers. *SIAM J. Matrix Anal. Appl.*, 13(1):138–144, 1992.

[2] R. Barrett, M. Berry, T. F. Chan, and et al. *Templates for the solution of linear systems: building blocks for iterative methods.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.

[3] C. H. Bischof. Incremental condition estimation. *SIAM J. Matrix Anal. Appl.*, 35(11):312–322, 1990.

[4] I. S. Duff and C. Vömel. Incremental norm estimation for dense and sparse matrices. *BIT*, 42(2):300–322, 2002.

[5] J. Duintjer Tebbens and M. Tůma. On incremental condition estimators in the 2-norm. *SIAM J. Matrix Anal. Appl.*, 35(1):174–197, 2014.

[6] G. H. Golub and G. Meurant. *Matrices, moments and quadrature with applications.* Princeton University Press, USA, 2010.

[7] G. H. Golub and C. F. Van Loan. *Matrix computations.* Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[8] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Research Nat. Bur. Standards*, 49:409–436, 1952.

[9] N. J. Higham. *Accuracy and stability of numerical algorithms.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996.

[10] G. Meurant. *The Lanczos and conjugate gradient algorithms, from theory to finite precision computations*, volume 19 of *Software, Environments, and Tools.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2006.

[11] G. Meurant. Estimates of the l_2 norm of the error in the conjugate gradient algorithm. *Numer. Algorithms*, 40(2):157–169, 2005.

[12] C. C. Paige and M. A. Saunders. LSQR: an algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Software*, 8(1):43–71, 1982.

[13] J.-L. Rigal and J. Gaches. On the compatibility of a given solution with the data of a linear system. *J. Assoc. Comput. Mach.*, 14:543–548, 1967.