SIAM J. MATRIX ANAL. APPL.
Vol. 40, No. 1, pp. 235–253

EUCLIDEAN-NORM ERROR BOUNDS FOR SYMMLQ AND CG*

RON ESTRIN†, DOMINIQUE ORBAN‡, AND MICHAEL SAUNDERS§

Abstract. For positive definite and semidefinite consistent $Ax_\star = b$, we use the Gauss–Radau approach of Golub and Meurant (1997) to obtain an upper bound on the error $\|x_\star - x_k^L\|_2$ for SYMMLQ iterates, assuming exact arithmetic. Such a bound, computable in constant time per iteration, was not previously available. We show that the CG error $\|x_\star - x_k^C\|_2$ is always smaller and can also be bounded in constant time per iteration. Our approach is computationally cheaper than other bounds or estimates of the CG error in the literature. As with other approaches using Gauss–Radau quadrature, we require a positive lower bound on the smallest nonzero eigenvalue of A. For indefinite A, we obtain an estimate of $\|x_\star - x_k^L\|_2$. Numerical experiments demonstrate that our bounds are remarkably tight for SYMMLQ on positive definite systems and therefore provide reliable bounds for CG.

Key words. symmetric linear equations, iterative method, Krylov subspace method, Lanczos process, CG, SYMMLQ, error estimates

AMS subject classifications. 65F10, 65F50

DOI. 10.1137/16M1094816

1. Introduction.

We consider the conjugate gradient method (CG) (Hestenes and Stiefel, 1952) and SYMMLQ (Paige and Saunders, 1975) for solving symmetric linear systems $Ax = b$, where $A \in \mathbb{R}^{n\times n}$ is a sparse symmetric matrix or a fast linear operator, i.e., one for which operator-vector products Av can be computed efficiently. For $x_0 = 0$, the kth iterates x_k^C and x_k^L formed by CG and SYMMLQ lie in the kth Krylov subspace $\mathcal{K}_k = \operatorname{span}\{b, Ab, \ldots, A^{k-1}b\}$. In exact arithmetic, Krylov methods ensure there is an iteration $\ell \le n$ for which $x_\ell^C = x_{\ell+1}^L = x_\star$, the pseudoinverse (min-length) solution, where x_k^L is defined for iterations $k = 2, \ldots, \ell+1$. (Our notation differs from that of Paige and Saunders (1975) so that both x_k^L and x_k^C are in $\mathcal{K}_k$.)

When A is positive definite, it is known that the CG error $\|x_\star - x_k^C\|_2$ is monotonic (Hestenes and Stiefel, 1952, Theorem 6:3), although it is not minimized in $\mathcal{K}_k$ at each iteration. The error is also monotonic for SYMMLQ, as it is minimized in a related space (Saunders, 2016). Empirically, CG typically maintains a smaller error than SYMMLQ by an order of magnitude, but neither CG nor SYMMLQ provides an obvious estimate of the error from above. Although the norm of the residual, $r = b - Ax = A(x_\star - x)$, can be computed, it may yield loose bounds that depend on the condition number of A, such as

$$\|x_\star - x\|_2 \le \|r\|_2\|A^{-1}\|_2 \quad \text{and} \quad \frac{\|x_\star - x\|_2}{\|x_\star\|_2} \le \frac{\|r\|_2}{\|b\|_2}\|A\|_2\|A^{-1}\|_2.$$

*Received by the editors September 21, 2016; accepted for publication (in revised form) November 13, 2018; published electronically February 14, 2019.
http://www.siam.org/journals/simax/40-1/M109481.html
Funding: The research of the second author was partially supported by an NSERC Discovery Grant. The research of the third author was partially supported by the National Institute of General Medical Sciences of the National Institutes of Health (award U01GM102098).
†Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305-4042 (restrin@stanford.edu).
‡GERAD and Department of Mathematics and Industrial Engineering, École Polytechnique, Montréal, QC H3C 3A7, Canada (dominique.orban@gerad.ca).
§Systems Optimization Laboratory, Department of Management Science and Engineering, Stanford University, Stanford, CA 94305-4121 (saunders@stanford.edu).

Tighter estimates of the CG error using Gauss–Radau quadrature are developed by Golub and Meurant (1997), Meurant (1997, 2005), and Frommer, Kahl, Lippert, and Rittich (2013).

Here, we derive cheaply computable estimates of the error for both CG and SYMMLQ. Our estimates are upper bounds when A is symmetric positive definite, or when A is symmetric positive semidefinite and the system is consistent. As with the other approaches using Gauss–Radau quadrature, we require a positive lower bound on the smallest nonzero eigenvalue of A.

In section 2 we provide a brief overview of SYMMLQ. In section 3 we derive upper bounds on the SYMMLQ and CG errors when A is positive semidefinite, the system is consistent, and under the assumption that computations are carried out in exact arithmetic. Section 4 gives recursions for the error bounds. In section 5 we discuss the implications when A is indefinite, and in section 6 we discuss parameter choices for the error estimates. In section 7 we compare our error bounds with existing bounds and estimates. We test the error estimates on problems from the SuiteSparse Matrix Collection and compare them against existing approaches in section 8. We discuss use of the error bounds in termination criteria in section 9. Note that our derivations assume exact computation. The numerical experiments suggest that the theoretical upper bounds remain upper bounds in practice until convergence if the eigenvalue estimate λ_{est} is reasonable. A finite-precision analysis is left for future work.

1.1. Notation. Matrices are denoted by capital letters A, $B, \dots$, vectors by lowercase letters v, $w, \dots$, and scalars by Greek letters α, β, $\gamma, \dots$, with exceptions for c and s, which are used for plane reflections with $c^2 + s^2 = 1$. We use e_k to denote column k of an identity matrix of appropriate size, $\|\cdot\|$ denotes the Euclidean norm, and $\|\cdot\|_A$ is the energy norm defined by $\|u\|_A^2 := u^T A u$ for A symmetric positive definite (SPD). If A is symmetric, $\lambda_{|\min|}(A)$ denotes its smallest eigenvalue in absolute value.

For brevity, we use the term "error" to refer to both the error vector and the norm of the error, depending on the context.

We assume that $x_0 = 0$. If a nonzero starting vector x_0 is available, we take "$Ax_\star = b$" to be $A\Delta x = b - Ax_0$ with a zero starting vector; then $x_\star = x_0 + \Delta x$.

2. Overview of CG and SYMMLQ. Both CG and SYMMLQ may be derived from the Lanczos (1950) process, which generates orthonormal vectors $v_k \in \mathcal{K}_\ell$ such that, at the kth iteration, we have the factorization

$$AV_k = V_k T_k + \beta_{k+1} v_{k+1} e_k^T = V_{k+1} \underline{T}_k, \tag{1}$$

where $V_k = [v_1 \dots v_k]$ is orthonormal in exact arithmetic,

$$T_k = \begin{bmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_k \\ & & \beta_k & \alpha_k \end{bmatrix} = \begin{bmatrix} T_{k-1} & \beta_k e_{k-1} \\ \beta_k e_{k-1}^T & \alpha_k \end{bmatrix}, \qquad \text{and} \qquad \underline{T}_k = \begin{bmatrix} T_k \\ \beta_{k+1} e_k^T \end{bmatrix}.$$

In particular, $\beta_1 v_1 = b$ with $\beta_1 := \|b\|$. The iterates $x_k^C = V_k y_k^C$ and $x_k^L = V_k y_k^L$ are defined by the following subproblems (Saunders, 1995):

$$T_k y_k^C = \beta_1 e_1 \qquad \text{and} \qquad y_k^L = \arg\min_{y \in \mathbb{R}^k} \|y\| \quad \text{such that} \quad \underline{T}_{k-1}^T y = \beta_1 e_1. \tag{2}$$

For reference, the CG iterates are defined by Hestenes and Stiefel (1952) as

$$x_k^C = \underset{x\in\mathcal{K}_k}{\arg\min}\, \|x_\star - x\|_A,$$

and the SYMMLQ points are characterized (Fischer, 1996; Saunders, 2016) by

$$\begin{aligned} x_k^L &= \underset{x\in\mathcal{K}_k}{\arg\min}\, \|x\| \quad \text{such that} \quad b - Ax \perp \mathcal{K}_{k-1} \\ &= \underset{x\in A\mathcal{K}_{k-1}}{\arg\min}\, \|x_\star - x\|, \quad \text{with} \quad A\mathcal{K}_{k-1} = \operatorname{span}\left\{Ab, A^2b, \dots, A^{k-1}b\right\}. \end{aligned}$$

When A is singular but $Ax = b$ is consistent, Krylov subspace methods identify the same (minimum-norm) solution, as explained in the following proposition.

Proposition 1. *Assume symmetric A is singular but $Ax = b$ is consistent. Let $x_\star$ be the solution produced by a Krylov subspace method for solving $Ax_\star = b$; that is, $x_\star \in \mathcal{K}_\ell$ for some ℓ. Then $x_\star$ is the unique solution to*

$$\min\ \|x\| \quad \textit{subject to} \quad Ax = b. \tag{3}$$

Proof. First note that necessary and sufficient conditions for $x_\star$ to solve (3) are that $Ax_\star = b$ and $x_\star \in \operatorname{range}(A)$. Since $Ax = b$ is consistent, $b \in \operatorname{range}(A)$, and so the Krylov subspace is contained in range(A), implying that $x_\star \in \mathcal{K}_k \subseteq \operatorname{range}(A)$. Since $Ax_\star = b$ and $x_\star \in \operatorname{range}(A)$, it must be the solution to (3). ∎

Proposition 1 implies that CG and SYMMLQ will identify the same solution to $Ax = b$.

2.1. The SYMMLQ iterates. We provide some key properties of SYMMLQ and describe some of the quantities that are computed at the kth iteration. Many of the factorizations are reused and modified to obtain estimates of the SYMMLQ and CG error. A more detailed treatment is given by Paige and Saunders (1975), from which we derive most of the notation (with minor differences).

To obtain x_k^L, we compute the LQ factorization $T_{k-1}Q_{k-1}^T = \overline{L}_{k-1}$, where Q_{k-1} is orthogonal and

$$\bar{L}_{k-1} = \begin{bmatrix} \gamma_1 & & & & \\ \delta_2 & \gamma_2 & & & \\ \varepsilon_3 & \delta_3 & \gamma_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \varepsilon_{k-1} & \delta_{k-1} & \bar{\gamma}_{k-1} \end{bmatrix}.$$

Note that the diagonal entries of $\bar{L}_{k-1}$ are γ_j for $j = 1, \dots, k-2$, and the last entry is $\bar{\gamma}_{k-1}$. A single 2×2 reflection is applied on the right to obtain $\underline{T}_{k-1}^T Q_k^T = [L_{k-1}\ 0]$, so that L_{k-1} differs from $\bar{L}_{k-1}$ only in the last diagonal entry, which becomes γ_{k-1}. The reflection is constructed so that

$$\begin{bmatrix} \bar{\gamma}_{k-1} & \beta_k \\ \bar{\delta}_k & \alpha_k \\ 0 & \beta_{k+1} \end{bmatrix} \begin{bmatrix} c_k & s_k \\ s_k & -c_k \end{bmatrix} = \begin{bmatrix} \gamma_{k-1} & 0 \\ \delta_k & \bar{\gamma}_k \\ \varepsilon_{k+1} & \bar{\delta}_{k+1} \end{bmatrix}.$$

The first iteration begins with $k = 2$ (because SYMMLQ iterates are defined only for $k \ge 2$), and $\bar{\gamma}_1 = \alpha_1$ and $\bar{\delta}_2 = \beta_2$. For $k \ge 2$, define $z_{k-1} = \begin{bmatrix}\zeta_1 & \dots & \zeta_{k-1}\end{bmatrix}^T$ as the solution to $L_{k-1}z_{k-1} = \beta_1 e_1$. Note that $y_k^L = Q_k^T \left[\begin{smallmatrix} z_{k-1} \\ 0 \end{smallmatrix}\right]$ solves (2), so that

$$x_k^L = V_k y_k^L = V_k Q_k^T \begin{bmatrix} z_{k-1} \\ 0 \end{bmatrix} = \overline{W}_k \begin{bmatrix} z_{k-1} \\ 0 \end{bmatrix} = W_{k-1} z_{k-1} \tag{4}$$

with the orthogonal matrix $\overline{W}_k = V_k Q_k^T = \begin{bmatrix} w_1 & \dots & w_{k-1} & \bar{w}_k \end{bmatrix} = \begin{bmatrix} W_{k-1} & \bar{w}_k \end{bmatrix}$.

Paige and Saunders (1975) establish the following results.

LEMMA 2. *The SYMMLQ iterates x_k^L satisfy the following properties:*

1. $x_k^L = x_{k-1}^L + \zeta_{k-1} w_{k-1} \in \mathcal{K}_k$, *with* $w_{k-1} \perp x_{k-1}^L$. *Furthermore,* $\|x_k^L\| = \|z_{k-1}\|$ *and is monotonically increasing.*
2. *Since x_k^L is updated along orthogonal directions,* $\|x_\star - x_k^L\|^2 = \|x_\star\|^2 - \|x_k^L\|^2$ *is monotonically decreasing.*
3. *It is possible to transfer to the CG iterate via the update* $x_k^C = x_k^L + \bar{\zeta}_k \bar{w}_k$, *where* $\bar{\zeta}_k = \zeta_k / c_{k+1}$ *and* $\bar{w}_k \perp \mathcal{K}_k$ *are byproducts of the SYMMLQ iteration. Note that* $\|x_k^C\|^2 = \|x_k^L\|^2 + \bar{\zeta}_k^2$.

3. Upper bounds on the error when A is semidefinite. In this section, we derive an upper bound on the error in SYMMLQ and build upon it to derive an upper bound for CG. As with other Gauss–Radau based approaches, we assume the availability of a nonzero underestimate to the smallest nonzero eigenvalue of A.

We assume that A is positive semidefinite with rank $r \le n$, but that $Ax = b$ is consistent. The situation where A is SPD is simply a special case. Let the spectrum of A be ordered as $0 = \lambda_n = \cdots = \lambda_{r+1} < \lambda_r \le \cdots \le \lambda_1$, and consider an underestimate of the smallest nonzero eigenvalue $\lambda_{\text{est}} \in (0, \lambda_r)$. Under the above assumption, SYMMLQ and CG identify the pseudoinverse solution $x_\star = A^\dagger b = \arg\min_x \{\|x\| \mid Ax = b\}$. The Rayleigh–Ritz theorem states that

$$\lambda_r = \min\{v^T A v \mid v \in \text{Range}(A),\ \|v\| = 1\}.$$

In addition, for any $u \in \mathbb{R}^k$ with $\|u\| = 1$, $V_k u \in \text{Range}(A)$ because each $v_i \in \text{Range}(A)$, and $\|V_k u\| = 1$. Then, each T_k is positive definite because $u^T T_k u = (V_k u)^T A (V_k u) \ge \lambda_r > 0$. Because each x_k^L and x_k^C lie in Range(A) by definition, the SYMMLQ and CG iterations occur as if they were applied to the symmetric and positive definite system consisting in the restriction of $Ax = b$ to Range(A).

3.1. Existing error estimates for Krylov subspace methods. There has been significant interest in estimating the A-norm of the CG error, the history of which is detailed by Strakoš and Tichý (2002). The Euclidean norm has received less attention as it is more difficult to estimate for CG, although it has been studied by Strakoš and Tichý (2002), Golub and Meurant (1997), Meurant (1997, 2005), and Frommer et al. (2013). Although estimates for the CG error are derived by Meurant (2005), they are not proved to be upper bounds, while those of Frommer et al. (2013) are upper bounds but can be more expensive in ill-conditioned cases in order to achieve improved accuracy (by increasing d in section 7). The only Euclidean-norm SYMMLQ error upper bounds we are aware of are those of Szyld and Widlund (1993), who provide a pessimistic geometric error decay rate.

The strategy behind estimating error norms is to recognize the error and related quantities as quadratic forms $r^T f(A) r$ evaluated at A for a certain function f (for example, $f(\xi) = \xi^{-2}$ and $r = b - Ax$) and seek estimates of this quadratic form. If $A = P \Lambda P^T$ is the eigenvalue decomposition of A, p_i is the ith column of P, and λ_i is the ith largest eigenvalue, then the quadratic form can be expressed as

$$b^T f(A) b := b^T P f(\Lambda) P^T b = \sum_{i=1}^{n} f(\lambda_i) \phi_i^2, \qquad \phi_i := p_i^T b,\ i = 1, \dots, n. \tag{5}$$

The connection between such quadratic forms and their approximation via Gaussian quadrature is most notably studied by Dahlquist, Eisenstat, and Golub (1972), Dahlquist, Golub, and Nash (1979), and Golub and Meurant (1994, 1997), who show it is possible to derive upper and lower bounds using the Lanczos process on (A, b). We follow this strategy to bound the SYMMLQ and CG errors.

3.2. Upper bounds on the SYMMLQ error. According to (4) and result 2 of Lemma 2, we have

$$\|x_\star - x_k^L\|^2 = \|x_\star\|^2 - \|x_k^L\|^2 = \|x_\star\|^2 - \|z_{k-1}\|^2. \tag{6}$$

Thus it is sufficient to find an upper bound on $\|x_\star\|^2 = b^T A^{-2} b$, assuming temporarily for the clarity of exposition that A is SPD. In this section, we show how to obtain such a bound at the cost of a few scalar operations per iteration.

We are interested in the choices $f(\xi) = \xi^{-2}$ (with $\xi = A$) as well as $f(\xi) = \xi^{-1}$ (with $\xi = A^2$). Although these appear to be exactly the same, the estimation procedure and convergence properties of the estimates are different when A is indefinite, since A^2 is guaranteed to be positive semidefinite.

When A is only semidefinite, we need to estimate the quadratic form $\|x_\star\|^2 = b^T \left(A^\dagger\right)^2 b = b^T f(A) b$, where

$$f(\xi) = \begin{cases} \xi^{-2}, & \xi > 0, \\ 0, & \xi = 0. \end{cases} \tag{7}$$

From the eigensystem $A = P \Lambda P^T$, this quadratic form is expressible as

$$\|x_\star\|^2 = \sum_{i=1}^{r} \lambda_i^{-2} \phi_i^2, \qquad \phi_i = p_i^T b, \; i = 1, \ldots, r.$$

Compared to (5), the only difference is that we now compute the sum over the nonzero eigenvalues.

We do not repeat the derivation of using Gauss–Radau quadrature to obtain an upper bound on such quadratic forms. The details can be found in (Golub and Meurant, 1994, 2009; Meurant, 2006). The following key theorem is the basis of our approach.

THEOREM 3. *Let A be positive semidefinite, $Ax = b$ be consistent, $f : (0, \infty) \to \mathbb{R}$, and the derivatives of f satisfy $f^{(2m+1)}(\xi) < 0$ for all $\xi \in (\lambda_r, \lambda_{\max}(A))$ and all integers $m \geq 0$. Fix $\lambda_{\mathrm{est}} \in (0, \lambda_r)$. Let T_k be generated by k steps of the Lanczos process on (A, b), and let*

$$\widetilde{T}_k := \begin{bmatrix} T_{k-1} & \beta_k e_{k-1} \\ \beta_k e_{k-1}^T & \omega_k \end{bmatrix},$$

where ω_k is chosen such that $\lambda_{\min}(\widetilde{T}_k) = \lambda_{\mathrm{est}}$. Then

$$b^T f(A) b \leq \|b\|^2 e_1^T f(\widetilde{T}_k) e_1.$$

Proof. The result follows from (Golub and Meurant, 1994, Theorem 3.2) and the section preceding it, as well as (Golub and Meurant, 1994, Theorem 3.4), although those results only consider the case where A is SPD. ∎

Because $T_{k-1} = V_{k-1}^T A V_{k-1}$ in exact arithmetic, the Poincaré separation theorem ensures that $\lambda_r \leq \lambda_{\min}(T_{k-1}) \leq \lambda_{\max}(T_{k-1}) \leq \lambda_{\max}(A)$ for all k. On the other hand, the Cauchy interlace theorem guarantees that $\lambda_{\min}(\widetilde{T}_k) < \lambda_{\min}(T_{k-1})$. As Theorem 3 announces, because $\lambda_r > 0$, it is possible to select ω_k to achieve a prescribed $\lambda_{\min}(\widetilde{T}_k)$.

The objective is to compute ω_k in $\widetilde{T}_k$ and then efficiently evaluate the quadratic form. Golub and Meurant (1994) show that $\omega_k = \lambda_{\text{est}} + \eta_{k-1}$, where η_{k-1} is obtained from the last entry of the solution of the system

$$(T_{k-1} - \lambda_{\text{est}} I)\, u_{k-1} = \beta_k^2 e_{k-1}. \tag{8}$$

To compute u_{k-1}, we take the QR factorization of $T_{k-1} - \lambda_{\text{est}} I$ analogous to the LQ factorization of $\underline{T}_{k-1}^T$ in SYMMLQ. This differs from (Orban and Arioli, 2017), where a Cholesky factorization is used, but QR factorization allows us to solve the indefinite system using a stable factorization. It begins with the 2×2 reflection

$$\begin{bmatrix} c_1^{(\omega)} & s_1^{(\omega)} \\ s_1^{(\omega)} & -c_1^{(\omega)} \end{bmatrix} \begin{bmatrix} \alpha_1 - \lambda_{\text{est}} & \beta_2 & \\ \beta_2 & \alpha_2 - \lambda_{\text{est}} & \beta_3 \end{bmatrix} = \begin{bmatrix} \rho_1 & \sigma_2 & \tau_3 \\ & \bar\rho_2 & \bar\sigma_3 \end{bmatrix}$$

and proceeds with reflections defined by

$$\begin{bmatrix} c_j^{(\omega)} & s_j^{(\omega)} \\ s_j^{(\omega)} & -c_j^{(\omega)} \end{bmatrix} \begin{bmatrix} \bar\rho_j & \bar\sigma_{j+1} & \\ \beta_{j+1} & \alpha_{j+1} - \lambda_{\text{est}} & \beta_{j+2} \end{bmatrix} = \begin{bmatrix} \rho_j & \sigma_{j+1} & \tau_{j+2} \\ & \bar\rho_{j+1} & \bar\sigma_{j+2} \end{bmatrix}.$$

Putting the QR factorization together, we have

$$T_{k-1} - \lambda_{\text{est}} I = \begin{bmatrix} \times & \times & \cdots & \times \\ \times & \times & & \times \\ & \ddots & \ddots & \vdots \\ & & s_{k-2}^{(\omega)} & -c_{k-2}^{(\omega)} \end{bmatrix} \begin{bmatrix} \rho_1 & \sigma_2 & \tau_3 & & \\ & \rho_2 & \sigma_3 & \ddots & \\ & & \rho_3 & \ddots & \tau_{k-1} \\ & & & \ddots & \sigma_{k-1} \\ & & & & \bar\rho_{k-1} \end{bmatrix},$$

where $\times$ is a placeholder for entries we are not interested in. We do not need to compute the QR factorization fully as we require only the scalars $s_{k-2}^{(\omega)}$, $c_{k-2}^{(\omega)}$, and $\bar\rho_{k-1}$ at the kth iteration. The relevant recurrence relations are

$$\begin{aligned} \bar\rho_1 &= \alpha_1 - \lambda_{\text{est}}, \\ \bar\sigma_2 &= \beta_2, \qquad c_0^{(\omega)} = -1, \\ \rho_1 &= \sqrt{\bar\rho_1^2 + \beta_2^2}, \qquad c_1^{(\omega)} = \frac{\alpha_1 - \lambda_{\text{est}}}{\rho_1}, \qquad s_1^{(\omega)} = \frac{\beta_2}{\rho_1}; \end{aligned}$$

for $k \ge 2$,

$$\begin{aligned} \bar\rho_k &= s_{k-1}^{(\omega)} \bar\sigma_k - c_{k-1}^{(\omega)} (\alpha_k - \lambda_{\text{est}}), \\ \bar\sigma_{k+1} &= -c_{k-1}^{(\omega)} \beta_{k+1}, \qquad \tau_k = s_{k-2}^{(\omega)} \beta_k, \\ \rho_k &= \sqrt{\bar\rho_k^2 + \beta_{k+1}^2}, \qquad c_k^{(\omega)} = \frac{\bar\rho_k}{\rho_k}, \qquad s_k^{(\omega)} = \frac{\beta_{k+1}}{\rho_k}. \end{aligned}$$

From the QR factorization of (8), we see that

$$\begin{bmatrix} \rho_1 & \sigma_2 & \tau_3 & & \\ & \rho_2 & \sigma_3 & \ddots & \\ & & \rho_3 & \ddots & \tau_{k-1} \\ & & & \ddots & \sigma_{k-1} \\ & & & & \bar\rho_{k-1} \end{bmatrix} \begin{bmatrix} \times \\ \vdots \\ \times \\ \eta_{k-1} \end{bmatrix} = \begin{bmatrix} \times & \times & & \\ \times & \times & \ddots & \\ \vdots & & \ddots & s_{k-2}^{(\omega)} \\ \times & \cdots & \cdots & -c_{k-2}^{(\omega)} \end{bmatrix} \beta_k^2 e_{k-1} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \beta_k^2 s_{k-2}^{(\omega)} \\ -\beta_k^2 c_{k-2}^{(\omega)} \end{bmatrix},$$

and therefore $\eta_{k-1} = -\beta_k^2 c_{k-2}^{(\omega)}/\bar{\rho}_{k-1}$, with $\omega_k = \lambda_{\text{est}} + \eta_{k-1}$.

We now describe how to compute $\beta_1^2 e_1^T \widetilde{T}_k^{-2} e_1$ efficiently. Note that if we take the LQ factorization of $\widetilde{T}_k = \widetilde{L}_k \widetilde{Q}_k$, then by symmetry of $\widetilde{T}_k$,

$$\begin{aligned} \beta_1^2 e_1^T \widetilde{T}_k^{-2} e_1 &= \beta_1^2 e_1^T \big(\widetilde{L}_k \widetilde{Q}_k\big)^{-T} \big(\widetilde{L}_k \widetilde{Q}_k\big)^{-1} e_1 \\ &= \beta_1^2 e_1^T \widetilde{L}_k^{-T} \widetilde{L}_k^{-1} e_1 = \|\beta_1 \widetilde{L}_k^{-1} e_1\|^2 \\ &= \|\widetilde{z}_k\|^2, \end{aligned} \tag{9}$$

where $\widetilde{L}_k \widetilde{z}_k = \beta_1 e_1$. Because $\widetilde{T}_k$ differs from T_k only in the (k,k) entry, we have

$$\widetilde{L}_k = \begin{bmatrix} L_{k-1} & 0 \\ \varepsilon_k e_{k-2}^T + \psi_k e_{k-1}^T & \bar{\omega}_k \end{bmatrix}, \quad \text{where} \quad \begin{bmatrix} c_k & s_k \\ s_k & -c_k \end{bmatrix} \begin{bmatrix} \bar{\delta}_k \\ \omega_k \end{bmatrix} = \begin{bmatrix} \psi_k \\ \bar{\omega}_k \end{bmatrix},$$

where ε_k comes from the LQ factorization of T_k. The vector $\widetilde{z}_k$ is closely related to z_k. Indeed, $L_{k-1} z_{k-1} = \beta_1 e_1$, and therefore

$$\widetilde{z}_k = \begin{bmatrix} z_{k-1} \\ \widetilde{\zeta}_k \end{bmatrix}, \qquad \widetilde{\zeta}_k = -\frac{1}{\bar{\omega}_k} \left(\varepsilon_k \zeta_{k-2} + \psi_k \zeta_{k-1}\right). \tag{10}$$

Theorem 3 (with f defined in (7)) and (9) imply that $\|x_\star\|^2 \le \|\widetilde{z}_k\|^2$ so that (6) yields

$$\|x_\star - x_k^L\|^2 = \|x_\star\|^2 - \|x_k^L\|^2 \le \|\widetilde{z}_k\|^2 - \|z_{k-1}\|^2 = (\epsilon_k^L)^2, \tag{11}$$

where we define

$$\epsilon_k^L := |\widetilde{\zeta}_k|. \tag{12}$$

Thus, with only a few extra floating-point operations per iteration we can compute an upper bound ϵ_k^L on the SYMMLQ error in the Euclidean norm.

Note that this approach can be applied when a positive definite preconditioner $M \approx A$ is used. The preconditioner changes the Lanczos decomposition, but all remaining computations carry through as above. We obtain an estimate of the error in the norm defined by the preconditioner, namely $\|x_\star - x_k\|_M$.

3.3. Upper bounds on the CG error. We now use the error bound derived in the previous section to obtain an upper bound on the CG error in the Euclidean norm. We first establish that the CG error is always lower than that of SYMMLQ for A positive semidefinite and $Ax = b$ consistent. Although the result yields the trivial upper bound (12), it also allows us to identify an improved bound. Define the kth CG direction as p_k with step length $\alpha_k^C > 0$, so that $x_k^C = \sum_{j=1}^k \alpha_j^C p_j$.

LEMMA 4 (Hestenes and Stiefel, 1952, Theorem 5:3). *The CG search directions satisfy* $p_i^T p_j \ge 0$ *for all* i, j.

The following lemma is also useful in our analysis.

LEMMA 5. *For* $1 \le k \le \ell$ *and* $0 \le d_1 \le d_2 \le \ell - k$,

$$(x_{k+d_2}^C)^T x_k^C \ge (x_{k+d_1}^C)^T x_k^C \ge \|x_k^C\|^2, \quad \textit{and, in particular,} \quad x_\star^T x_k^C \ge \|x_k^C\|^2.$$

Proof. Because $\alpha_i^C > 0$, Lemma 4 yields

$$\begin{aligned}(x_{k+d_2})^T x_k^C = \left(x_k^C + \sum_{i=k+1}^{k+d_2} \alpha_i^C p_i\right)^T x_k^C &= \left\|x_k^C\right\|^2 + \sum_{i=k+1}^{k+d_2}\sum_{j=1}^{k} \alpha_i^C \alpha_j^C p_i^T p_j \\ &\geq \left\|x_k^C\right\|^2 + \sum_{i=k+1}^{k+d_1}\sum_{j=1}^{k} \alpha_i^C \alpha_j^C p_i^T p_j \\ &\geq \left\|x_k^C\right\|^2. \qquad (13)\end{aligned}$$ ◻

We now relate the Euclidean-norm errors of SYMMLQ and CG.

THEOREM 6. *Let A be positive semidefinite and $Ax = b$ be consistent, and let $x_\star$ be the solution identified by both CG and SYMMLQ by virtue of Proposition 1. The following hold in exact arithmetic for all $2 \leq k \leq \ell$:*

$$\|x_k^L\| \leq \|x_k^C\|, \qquad (14)$$

$$\|x_\star - x_k^C\| \leq \|x_\star - x_k^L\|. \qquad (15)$$

Proof. Result 3 of Lemma 2 proves (14), and this with Lemma 5 implies

$$\left\|x_k^L\right\|^2 + \left\|x_k^C\right\|^2 \leq 2\left\|x_k^C\right\|^2 \leq 2x_\star^T x_k^C.$$

Rearranging and adding $\|x_\star\|^2$ to both sides gives

$$\|x_\star\|^2 - 2x_\star^T x_k^C + \left\|x_k^C\right\|^2 \leq \|x_\star\|^2 - \left\|x_k^L\right\|^2.$$

By factoring the left and using result 2 of Lemma 2 on the right, we obtain (15). ◻

Although the proof of Theorem 6 assumes exact arithmetic, we have observed empirically that the result holds until the error in x_k^L plateaus at convergence.

Theorem 6 immediately establishes the trivial bound

$$\left\|x_\star - x_k^C\right\| \leq \left\|x_\star - x_k^L\right\| \leq \epsilon_k^L, \qquad (16)$$

which provides an upper bound on the Euclidean-norm CG error, in contrast to the estimates of Meurant (2005). We can improve bound (16) using a few observations.

From Lemma 5,

$$\theta_k := x_\star^T x_k^C - \|x_k^C\|^2 \geq 0. \qquad (17)$$

Hence from part 3 of Lemma 2

$$\begin{aligned}\left\|x_\star - x_k^C\right\|^2 &= \|x_\star\|^2 - 2x_\star^T x_k^C + \left\|x_k^C\right\|^2 \\ &= \|x_\star\|^2 - 2\theta_k - \left\|x_k^C\right\|^2 \\ &= \|x_\star\|^2 - 2\theta_k - \left\|x_k^L\right\|^2 - \bar{\zeta}_k^2,\end{aligned}$$

and since $\left\|x_\star - x_k^L\right\| \leq \epsilon_k^L = |\widetilde{\zeta}_k|$ it follows that

$$\begin{aligned}\left\|x_\star - x_k^C\right\|^2 &= \left\|x_\star - x_k^L\right\|^2 - \bar{\zeta}_k^2 - 2\theta_k \\ &\leq \widetilde{\zeta}_k^2 - \bar{\zeta}_k^2 - 2\theta_k \qquad (18) \\ &\leq \widetilde{\zeta}_k^2 - \bar{\zeta}_k^2. \qquad (19)\end{aligned}$$

Since $\bar{\zeta}_k$ is readily available as part of the SYMMLQ iteration, (19) is an improvement on the bound (16). Unfortunately, bound (18) is not computable because $x_\star$ is unavailable. We define

$$\epsilon_k^C := \sqrt{\widetilde{\zeta}_k^2 - \bar{\zeta}_k^2} \le \left|\widetilde{\zeta}_k\right| = \epsilon_k^L \tag{20}$$

as an upper bound on the error of the kth CG iterate.

From (13), we could further improve the error estimate by approximating θ_k from below by introducing a delay, implemented using the sliding-window approach originally appearing in Golub and Strakǒs (1994) (stabilized by Golub and Meurant (1997) and used by Meurant (2005) and Orban and Arioli (2017)). Given Lemma 5, we define an approximation of (17) as

$$\theta_k^{(d)} := (x_{k+d}^C)^T x_k^C - \|x_k^C\|^2 \le \theta_k \qquad (d > 0),$$

noting that $0 \le \theta_k^{(1)} \le \cdots \le \theta_k^{(\ell-k)} = \theta_k$.

We now describe how to compute $\theta_k^{(d)}$ without storing the iterates $x_k^C, \dots, x_{k+d}^C$ explicitly. Recalling that $x_k^C = x_k^L + \bar{\zeta}_k \bar{w}_k = \sum_{i=1}^{k-1} \zeta_i w_i + \bar{\zeta}_k \bar{w}_k$, we have

$$\begin{aligned}
\theta_k^{(d)} &= \left(x_k^L + \bar{\zeta}_k \bar{w}_k\right)^T \left(x_{k+d}^L + \bar{\zeta}_{k+d} \bar{w}_{k+d}\right) - \left(\|x_k^L\|^2 + \bar{\zeta}_k^2\right) \\
&= \|x_k^L\|^2 + \bar{\zeta}_k \bar{w}_k^T x_{k+d}^L + \bar{\zeta}_k \bar{\zeta}_{k+d} \bar{w}_k^T \bar{w}_{k+d} - \left(\|x_k^L\|^2 + \bar{\zeta}_k^2\right) \\
&= \bar{\zeta}_k \sum_{i=k}^{k+d-1} \zeta_i \bar{w}_k^T w_i + \bar{\zeta}_k \bar{\zeta}_{k+d} \bar{w}_k^T \bar{w}_{k+d} - \bar{\zeta}_k^2,
\end{aligned}$$

where we use the fact that $w_i^T w_j = 0$ for $i \ne j$ and $\bar{w}_i^T w_j = 0$ for $j < i$. We now use the fact that

$$\bar{w}_k^T w_i = c_{i+1} \prod_{j=k+1}^{i} s_j \quad \text{and} \quad \bar{w}_k^T \bar{w}_i = \prod_{j=k+1}^{i} s_j \quad \text{for } i \ge k,$$

so that

$$\theta_k^{(d)} = \bar{\zeta}_k \sum_{i=k}^{k+d-1} \left(\zeta_i c_{i+1} \prod_{j=k+1}^{i} s_j \right) + \bar{\zeta}_k \bar{\zeta}_{k+d} \prod_{j=k+1}^{k+d} s_j - \bar{\zeta}_k^2.$$

We can compute $\theta_k^{(d)}$ in $O(d)$ flops and $O(d)$ storage by maintaining d partial products of the form $\prod_{j=k+1}^{i} s_j$ for $k+1 \le i \le k+d$. At the next iteration we can divide each partial product by s_{k+1} and multiply the last one by s_{k+d} to obtain the necessary partial products for iteration $k+1$.

With the above expression we can improve (19) to

$$\|x_\star - x_k^C\|^2 \le \left(\epsilon_k^C\right)^2 - 2\theta_k^{(d)}. \tag{21}$$

This improved bound is only noticeable when λ_{est} is a close estimate to $\lambda_{\min}$. Otherwise, the difference between the ϵ_k^C and $\|x_\star - x_k^C\|$ is dominated by the error in the Gauss–Radau quadrature (the difference between ϵ_k^L and $\|x_\star - x_k^L\|$).

It is not necessary to implement CG via the transfer point from SYMMLQ in order to compute these error bounds because only $\{\alpha_k, \beta_k\}$ from the Lanczos process

Algorithm 1. SYMMLQ with CG error estimation.

Input: A, b, and λ_{est} such that $\lambda_{\text{est}} < \lambda_{\min}(A)$.
Obtain $\alpha_1, \beta_1, \beta_2$ of Lanczos process on (A, b)
$\bar{\gamma}_1 = \alpha_1$, $\bar{\delta}_2 = \beta_2$, $\varepsilon_1 = \varepsilon_2 = 0$ ▷ begin QR of $\bar{L}_k$
$\bar{\rho}_1 = \alpha_1 - \lambda_{\text{est}}$, $\bar{\sigma}_2 = \beta_2$, $\rho_1 = \sqrt{\bar{\rho}_1^2 + \beta_2^2}$ ▷ begin QR of (8)
$c_0^{(\omega)} = -1$, $c_1^{(\omega)} = (\alpha_1 - \lambda_{\text{est}})/\rho_1$, $s_1^{(\omega)} = \beta_2/\rho_1$
$\zeta_0 = 0$, $\bar{\zeta}_1 = \beta_1/\bar{\gamma}_1$ ▷ initialize remaining variables
for $k = 2, 3, \dots$ **do**
  $\gamma_{k-1} = \sqrt{\bar{\gamma}_{k-1}^2 + \beta_k^2}$
  $c_k = \bar{\gamma}_{k-1}/\gamma_{k-1}$, $s_k = \beta_k/\gamma_{k-1}$
  Obtain α_k, β_{k+1} from Lanczos process on (A, b)
  $\delta_k = \bar{\delta}_k c_k + \alpha_k s_k$, $\bar{\gamma}_k = \bar{\delta}_k s_k - \alpha_k c_k$ ▷ continue QR of $\bar{L}_k$
  $\varepsilon_{k+1} = \beta_{k+1} s_k$, $\bar{\delta}_{k+1} = -\beta_{k+1} c_k$
  $\zeta_{k-1} = \bar{\zeta}_{k-1} c_k$ ▷ forward substitution
  $\bar{\zeta}_k = -(\varepsilon_k \zeta_{k-2} + \delta_k \zeta_{k-1})/\bar{\gamma}_k$
  $\eta_{k-1} = -\beta_k^2 c_{k-2}^{(\omega)}/\bar{\rho}_{k-1}$ ▷ forward substitution on (8)
  $\omega_k = \lambda_{\text{est}} + \eta_{k-1}$
  $\psi_k = c_k \bar{\delta}_k + s_k \omega_k$, $\bar{\omega}_k = s_k \bar{\delta}_k - c_k \omega_k$
  $\epsilon_k^L = |(\varepsilon_k \zeta_{k-2} + \psi_k \zeta_{k-1})/\bar{\omega}_k|$ ▷ compute error bounds
  $\epsilon_k^C = \left((\epsilon_k^L)^2 - \bar{\zeta}_k^2\right)^{\frac{1}{2}}$
  $\bar{\rho}_k = s_{k-1}^{(\omega)} \bar{\sigma}_k - c_{k-1}^{(\omega)}(\alpha_k - \lambda_{\text{est}})$ ▷ continue QR of (8)
  $\bar{\sigma}_{k+1} = -c_{k-1}^{(\omega)} \beta_{k+1}$, $\rho_k = \sqrt{\bar{\rho}_k^2 + \beta_{k+1}^2}$
  $c_k^{(\omega)} = \bar{\rho}_k/\rho_k$, $s_k^{(\omega)} = \beta_{k+1}/\rho_k$
end for

are required. These can be recovered from the classic Hestenes and Stiefel (1952) implementation of CG using equations provided by Meurant (2005).

For positive semidefinite A, we have derived upper bounds on the SYMMLQ and CG errors when $Ax = b$ is consistent. Only a few extra scalar operations are needed per iteration, and $O(1)$ extra memory.

4. Complete algorithm. Algorithm 1 provides the complete algorithm to compute the error bounds ϵ_k^L and ϵ_k^C, given $\{\alpha_k, \beta_k\}$ from the Lanczos process. Although it did not make a difference in our numerical experiments, it may be safer in practice to compute reflections using a variant of (Golub and Van Loan, 2013, sect. 5.1.8).

5. Estimation of $\|x_\star - x_k^L\|$ with A indefinite. We now focus on the SYMMLQ error when A is indefinite. Theorem 3 no longer applies, and so $\beta_1^2 e_1^T \widetilde{T}_k^{-2} e_1$ is only an estimate of $\|x_\star\|$ rather than an upper bound.

There are two approaches. The first is to continue as in subsection 3.2 and accept ϵ_k^L as an estimate of the error rather than an upper bound. Alternatively we can treat $\|x_\star\|^2 = b^T (A^2)^\dagger b$ as a quadratic form in A^2 rather than A. (Recall that for real symmetric A, $(A^2)^\dagger = (A^\dagger)^2$.) We formulate the problem as upper bounding the energy norm $\|x_\star\| = \|b\|_{B^\dagger}$ with $B = A^2$. Such computation is akin to computing the energy norm error for CG using Gauss–Radau quadrature, which has been studied by Golub and Meurant (1997) and others. The main difficulty is that it requires applying the Lanczos process to A^2 and b, which means two applications of A per iteration of

SYMMLQ. Although this theoretically guarantees that we obtain an upper bound on $\|x_\star\|$ (and therefore an upper bound on the error), roundoff error can diminish the quality of the estimation.

With these ideas in mind, we consider the procedure outlined in subsection 3.2, treating $b^T(A^2)^\dagger b$ as a quadratic form in A to estimate the error. In numerical experiments we observe that the estimate often remains an upper bound, even as the iterates converge to the solution. It is possible to loosen the error estimate by choosing a smaller value for λ_{est} to encourage the estimate to remain an upper bound; however, without knowing $\lambda_{|\min|}$, this may not be a practical solution. This is also illustrated in the numerical experiments.

Note that with A indefinite, λ_{est} should be chosen between zero and the eigenvalue closest to zero (keeping the sign of that eigenvalue). This is the only difference in the computation of ϵ_k^L. There may be iterations where $T_{k-1} - \lambda_{\text{est}} I$ becomes singular, and it may not be possible to compute ϵ_k^L for that iteration, but the QR factorization of $T_k - \lambda_{\text{est}} I$ will remain computable at future iterations.

6. The choice of $\boldsymbol{\lambda_{\text{est}}}$. A reasonably tight underestimate of λ_{est} is required for approaches using Gauss–Radau quadrature, such as for the error estimates proposed by Meurant (1997) and Frommer et al. (2013). The quality of our error bounds is directly dependent on the quality of the Gauss–Radau quadrature, which in turn depends on the quality of the eigenvalue estimate. Meurant and Tichý (2015) investigated the effect of λ_{est} on the quality of Gauss–Radau quadrature for the CG A-norm error.

If $\lambda_{|\min|} := \arg\min_{\lambda \in \Lambda(A)} |\lambda|$ is known, one should choose $\lambda_{\text{est}} = (1-\epsilon)\lambda_{|\min|}$ with $\epsilon \ll 1$. In the experiments below, we usually use $\epsilon = 10^{-10}$. Choosing λ_{est} slightly closer to zero alleviates numerical stability issues in computing ω_k with a near-singular $T_k - \lambda_{\text{est}} I$. This also applies when A is indefinite.

One example where it is trivial to obtain an underestimate of the smallest eigenvalue is for shifted linear systems $(A + \delta I)x = b$ with A SPD and $\delta > 0$, where the choice $\lambda_{\text{est}} = \delta$ may give good error estimates if A is close to singularity. This is of interest for regularized least-squares problems $(A^TA + \delta^2 I)x = A^Tb$ and is exploited by Estrin, Orban, and Saunders (2016).

When $\lambda_{|\min|}$ is not known, the choice of λ_{est} becomes application-specific. It may be possible to estimate the smallest eigenvalue as the iterations progress, similarly to Frommer et al. (2013), although this is the subject of ongoing research. If no information is known about the spectrum of A, Gauss–Radau quadrature approaches such as the one presented in this paper may not be practical.

7. Previous error estimates. As discussed in subsection 3.1, there are other approaches to estimating the error in the iterates of Krylov subspace methods, particularly for CG. In this section we provide a brief overview of the approaches taken by Brezinski (1999), Meurant (2005), and Frommer et al. (2013) as applied to CG, followed by some numerical experiments comparing the approaches. Only the error estimate by Brezinski (1999) applies to SYMMLQ as well. We include this in the numerical experiments.

Brezinski (1999) describes several estimates of the error for nonsingular square systems, including

$$\|x_\star - x_k\| \approx \frac{\|r_k\|^2}{\|Ar_k\|}, \qquad r_k = b - Ax_k \tag{22}$$

Table 1

*Cost of computing an error estimate for CG using various methods, where d is the window size for methods using a delay (denoted by *). The right column refers to whether the method guarantees an upper bound in exact arithmetic.*

	Cost per iteration	Storage	Upper bound
Brezinski (1999)	$O(n + nnz(A))$	$O(1)$	Yes, if scaled by $\kappa(A)$
Meurant (2005)*	$O(1)$	$O(d)$	No
Frommer et al. (2013)*	$O(d^2)$	$O(d)$	Yes
This paper, bound (20)	$O(1)$	$O(1)$	Yes
This paper, bound (21)*	$O(d)$	$O(d)$	Yes

(see also Auchmuty (1992)). This estimate is simple to implement, but requires an extra product Ar_k at each iteration. The estimate can be made into an upper bound by multiplying it by the condition number of A, or an upper bound thereof, assuming the latter is known ahead of time, although this considerably loosens the estimate. Thus, such conversion to an upper bound is only possible when A is nonsingular.

Meurant (2005) uses the relation

$$\|x_\star - x_k^C\|^2 = \|b\|^2 \left(e_1^T T_n^{-2} e_1 - e_1^T T_k^{-2} e_1\right) + (-1)^k \beta_{k+1} \|x_\star - x_k^C\|_A^2 \frac{\|b\|}{\|r_k^C\|} e_k^T T_k^{-2} e_1 \tag{23}$$

to relate the A-norm error to that of the Euclidean error for CG iterates. The first term can be approximated by introducing a delay d and replacing $e_1^T T_n^{-2} e_1$ by $e_1^T T_{k+d}^{-2} e_1$. The A-norm error can be estimated via Gauss quadrature as described by Golub and Meurant (1997), and the remaining terms by updating a QR factorization of T_k, so that the total cost is only $O(1)$ flops per iteration.

Frommer et al. (2013) use the fact that $r_k^C = \|r_k^C\| v_{k+1}$, where v_{k+1} is the $(k+1)$th Lanczos vector, and so

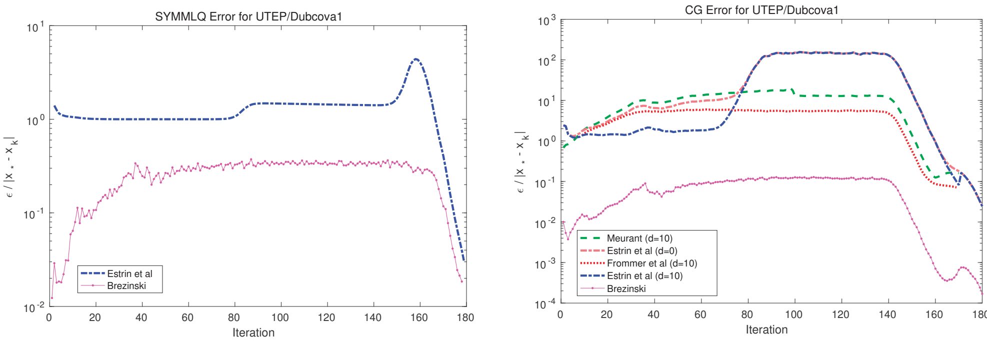
$$\|x_\star - x_k^C\|^2 = \|r_k^C\|^2 v_{k+1}^T A^{-2} v_{k+1}. \tag{24}$$

The right-hand side of (24) is upper-bounded using Gauss–Radau quadrature. Rather than restarting the Lanczos process on A using v_{k+1} as the initial vector at each CG iteration, they cleverly perform the Lanczos process on the lower $2d \times 2d$ submatrix of T_{k+d+1} using e_{d+1} as the starting vector, thus recovering the same estimate. The restarted Lanczos factorization requires $O(d^2)$ flops at each iteration.

In Table 1 we summarize the costs of the various error estimates for CG and say whether the estimate can be shown to be an upper bound in exact arithmetic.

8. Numerical experiments.

8.1. Comparison with previous estimates. We give some numerical examples comparing the various error estimation procedures for CG and SYMMLQ, using SPD matrices from the SuiteSparse Matrix Collection (Davis and Hu, 2011) and MATLAB implementations of all error estimates described in section 7. In each experiment, we use $b = \mathbb{1}/\sqrt{n}$ and compute $x_\star = A \backslash b$ via MATLAB. The solvers terminate when $\|r_k\| / \|b\| \le 10^{-10}$. For estimates using a delay d, we report the estimated error at iteration k using information obtained during iterations $k, k+1, \ldots, k+d$. Estimates requiring bounds on eigenvalues use $(1 - 10^{-10})\lambda_{\min}(A)$ for the lower bound and $(1 + 10^{-10})\lambda_{\max}(A)$ for the upper bound. (Further experiments in subsection 8.2 use a less accurate estimate of $\lambda_{\min}(A)$.) For each approach to estimating the error, we plot $\epsilon/\|x_* - x_k\|$, that is, the ratio of the estimate, ϵ, to the true error.



(a) SYMMLQ (b) CG. Window size $d = 10$ used.

FIG. 1. $\epsilon_k / \|x_\star - x_k\|$ *for SPD system UTEP/Dubcova*1 *using SYMMLQ and CG, where* ϵ_k *is the error bound for either SYMMLQ or CG.*

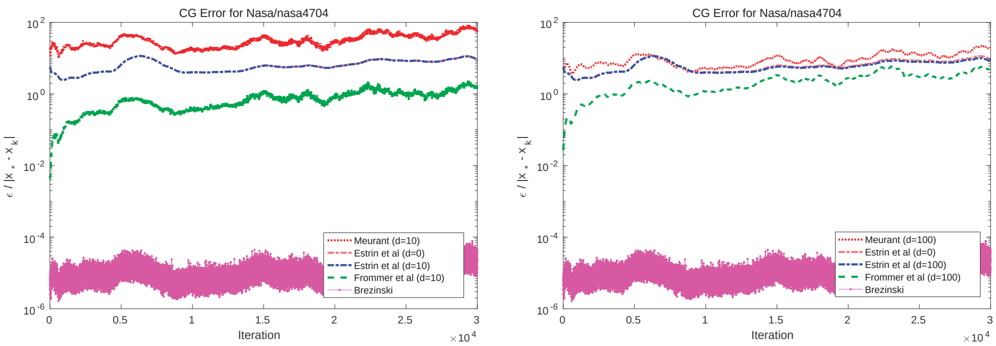
First we compare our SYMMLQ error estimate with that of Brezinski (1999). We use the matrix UTEP/Dubcova1 ($n = 16,129$ and $\kappa(A) \approx 10^3$). The ratio of the true error to the corresponding bounds is plotted in Figure 1a. We see that our bound is close to the true error until x_k^L attains its maximum accuracy, whereas the Brezinski (1999) estimate is a lower bound on the error for the examples in this section; however, if it is scaled by $\kappa(A)$, then it becomes a loose upper bound.

We now compare the estimates for CG from (20) and (21) using a well-conditioned system (again UTEP/Dubcova1) and an ill-conditioned system (Nasa/nasa4704, $n = 4704$ and $\kappa(A) \approx 10^7$). In Figure 1b, we see that all estimates do fairly well, as they are off by at most one or two orders of magnitude. Estimate (20) performs nearly as well as those of Meurant (2005) and Frommer et al. (2013) when $d = 10$ until a divergence occurs near iteration 70. The improved estimate (21) appears tightest until that same divergence occurs.

Next, we compare against the estimates of Meurant (2005) and Frommer et al. (2013) on Nasa/nasa4704 using $d = 10$ in Figure 2a and $d = 100$ in Figure 2b. We see that for $d = 10$, the (Meurant, 2005) estimate is not an upper bound, while that of Frommer et al. (2013) is looser than ours. The situation is improved for the other estimates with $d = 100$, where (20) and those of (Meurant, 2005; Frommer et al., 2013) are fairly similar, but the Meurant (2005) estimate is still not an upper bound, and the estimate of Frommer et al. (2013) is more costly for such d. We also note that in this case, increasing d does not noticeably improve (21) compared to (20).

For CG, (20) is the cheapest and in exact arithmetic is guaranteed to be an upper bound. At the same time, it is not necessarily the tightest estimate, and the estimate of Frommer et al. (2013) has the advantage of improved accuracy of the error estimate with increased window size d (more so than (21)), although at a higher computational cost, and it requires computing d iterations into the future. In some cases, such as Figure 2a, a good estimate that is not guaranteed to be a bound may be more useful, but without accuracy guarantees it may be difficult to use such estimates within termination criteria.

8.2. Additional SPD experiments. We evaluate the quality of our error bounds (12), (20), and (21) on further SPD examples from the SuiteSparse Collection. Again we solve $Ax = b$ with $b = \mathbb{1}/\sqrt{n}$, taking $x_\star = A \setminus b$ from MATLAB and terminating when $\|r_k\| / \|b\| \leq 10^{-10}$. We compute $\lambda_{|\min|}(A)$, the eigenvalue closest to zero,



(a) $d = 10$ (b) $d = 100$

FIG. 2. $\epsilon_k^C / \|x_\star - x_k^C\|$ *for SPD system Nasa/nasa*4704*. Delays* $d = 10$ *and* 100 *are used for estimates that take advantage of them.*

and obtain the error bounds using $\lambda_{\text{est}} = \mu\lambda_{|\min|}(A)$, typically with $\mu = 1 - 10^{-10}$ or 0.1. We also include a lower-bound error estimate using a delay (Hestenes and Stiefel, 1952; Golub and Strakŏs, 1994). Because SYMMLQ takes orthogonal steps,

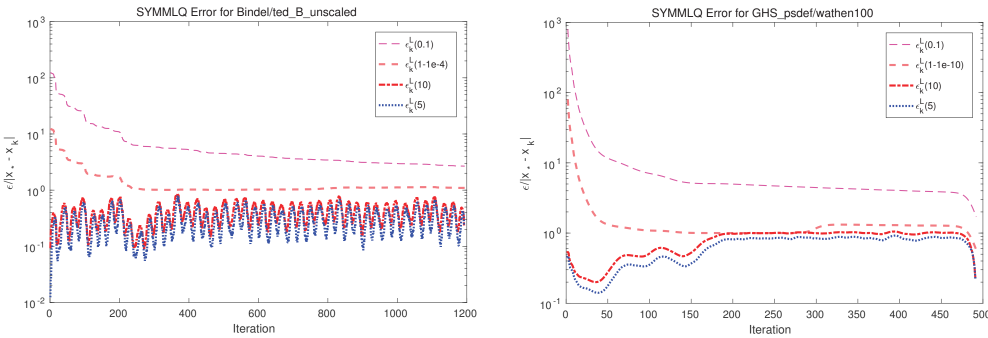
$$\left\|x_{k+d}^L - x_k^L\right\|^2 = \sum_{i=k}^{k+d-1} \zeta_i^2 \le \sum_{i=k}^{\ell} \zeta_i^2 = \left\|x_\star - x_k^L\right\|^2 \tag{25}$$

for any $d \ge 1$. By choosing a modest value $d = 5$ or 10 and storing the last d step lengths ζ_i, we can compute a lower bound on the error. Note that we can compute a lower bound via Gauss and Gauss–Radau quadrature with $\lambda_{\text{est}} \ge \|A\|_2$. Such techniques were used by Arioli (2013) and provide lower bounds comparable to those using a delay. We plot $\epsilon/\|x_\star - x_k\|$ to investigate the tightness of the bounds.

In the figure legends, $\epsilon_k^L(\mu)$ and $\epsilon_k^C(\mu)$ denote the error bounds for SYMMLQ and CG obtained from Gauss–Radau quadrature when $\lambda_{\text{est}} = \mu\lambda_{|\min|}(A)$, where $0 < \mu < 1$. For SYMMLQ we include the lower-bound error obtained using a delay with $d > 1$, denoted by $\epsilon_k^L(d)$.

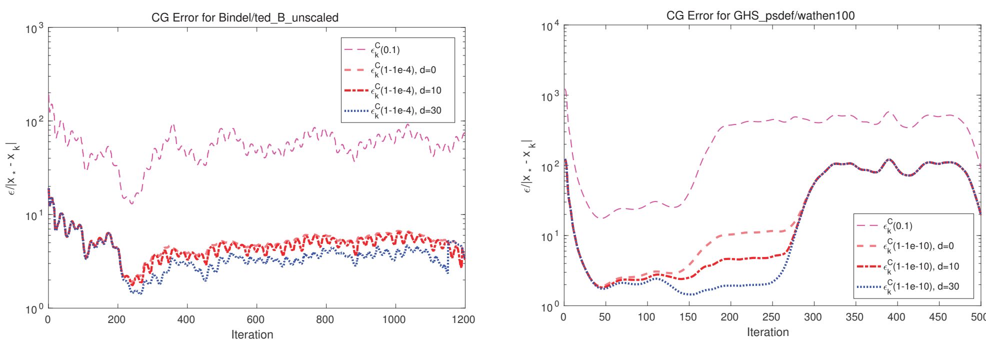
For SYMMLQ on Bindel/ted_B_unscaled ($n = 10605$ and $\kappa(A) \approx 10^{11}$), the bound to error ratios are shown in Figure 3a. For GHS_psdef/wathen100 ($n = 30401$ and $\kappa(A) \approx 10^3$), they are in Figure 3b. When λ_{est} approximates $\lambda_{|\min|} = \lambda_r$ well, the bound ϵ_k^L is remarkably tight after an initial lag. We used $\mu = 1 - 10^{-6}$ for the first problem due to A being ill-conditioned ($\lambda_{|\min|} \approx 10^{-11}$) and $\mu = 1 - 10^{-10}$ for the second problem. Even when λ_{est} is a tenth of the true eigenvalue, it appears that the bound is at most an order of magnitude larger, still outlining the true error from above. Only near convergence, ϵ_k^L may no longer be a bound when the true error plateaus. Having the computed bound continue to decrease after convergence is a desirable property for termination criteria. The lower bounds $\epsilon_k^L(d)$ oscillate an order of magnitude below the true error in Figure 3a, but in Figure 3b, both upper and lower bounds soon approximate the true error to within a couple orders of magnitude.

We now solve the same problems using CG. Figure 4 shows that ϵ_k^C is a considerably looser bound on the CG error than ϵ_k^L is on the SYMMLQ error, although both remain true upper bounds until convergence. As with SYMMLQ, if the error stagnates at convergence, the "bound" may continue to decrease. We see that increasing d in (21) (when using an accurate estimate of the smallest eigenvalue) improves the bound when A is reasonably conditioned but does not have a large impact for ill-



(a) Bindel/ted_B_unscaled (b) GHS_psdef/wathen100

FIG. 3. $\epsilon_k^L(\cdot)/\left\|x_\star - x_k^L\right\|$ *for two SPD systems. The Gauss–Radau approach gives upper bounds, while the delay gives lower bounds.*



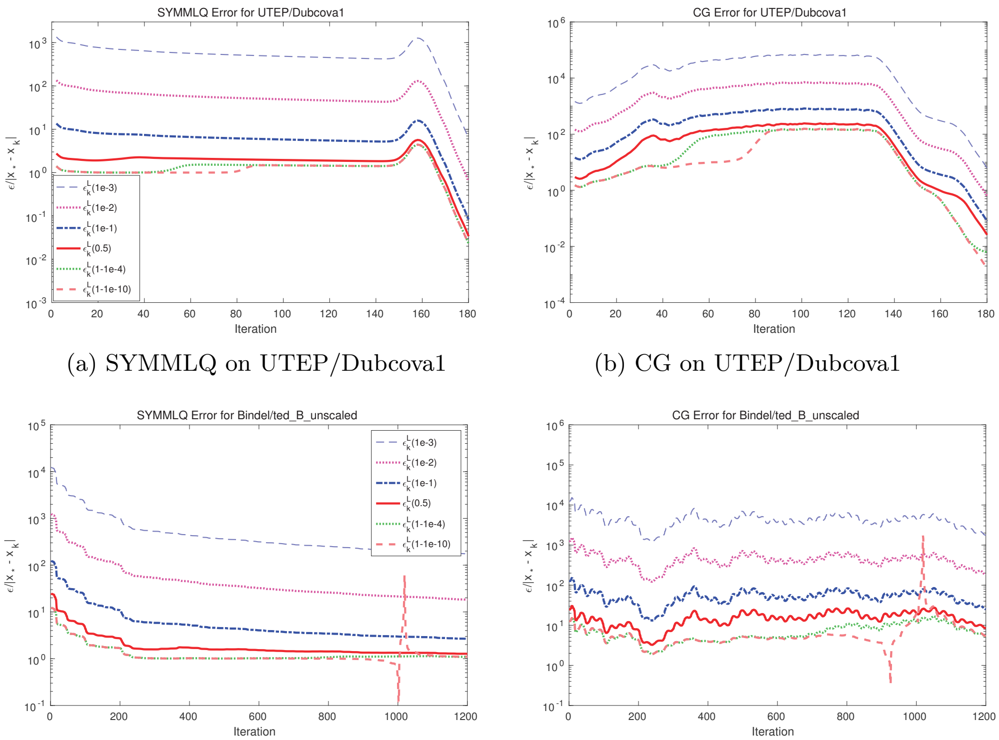
(a) Bindel/ted_B_unscaled (b) GHS_psdef/wathen100

FIG. 4. $\epsilon_k^C(\mu)/\left\|x_\star - x_k^C\right\|$ *for two SPD systems.*

conditioned problems. Also, ϵ_k^C diverges slightly from the true CG error when the error is roughly the square-root of the maximum attainable accuracy; in particular, d has nearly no noticeable effect past that point. This is probably due to $\bar{\zeta}_k$ becoming an order of magnitude smaller than ϵ_k^L.

8.3. Empirical check. To check whether the error bounds behave as upper bounds numerically, we ran SYMMLQ and CG on all SuiteSparse matrices of size $n \leq 25000$ with $\kappa(A) < 10^{16}$, resulting in 140 problems. We used $b = \mathbb{1}/\sqrt{n}$ and $\lambda_{\text{est}} = (1 - 10^{-10})\lambda_{\min}$ or $0.1\lambda_{\min}$ and terminated when the estimate $\epsilon_k^L, \epsilon_k^C \leq 10^{-10}$. We then counted the number of iterations where $\epsilon_k^L \geq \|x_\star - x_k^L\|$ and $\epsilon_k^C \geq \|x_\star - x_k^C\|$ were satisfied. For $\lambda_{\text{est}} = (1 - 10^{-10})\lambda_{\min}$ $(0.1\lambda_{\min})$, 121 (129) problems had ϵ_k^L and ϵ_k^C behave as upper bounds for all iterations, while for the remaining 19 (11) problems we saw a cross-over at convergence similar to Figure 3b, with ϵ_k^L and ϵ_k^C continuing to decrease once the true error plateaued. Thus empirically our bounds do behave as upper bounds until convergence.

8.4. Effect of $\boldsymbol{\lambda_{\text{est}}}$. We briefly investigate the effect of λ_{est} on the tightness of the error bounds (12) and (20). We use problems UTEP/Dubcova1 and Bindel/ted_B_unscaled again as examples of well- and ill-conditioned systems.



(a) SYMMLQ on UTEP/Dubcova1 (b) CG on UTEP/Dubcova1

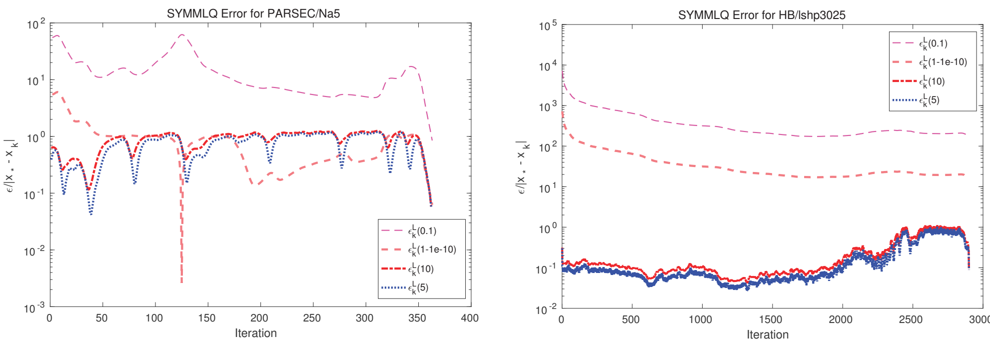
(c) SYMMLQ on Bindel/ted_B_unscaled (d) CG on Bindel/ted_B_unscaled

Fig. 5. $\epsilon_k(\mu)/\|x_k - x_\star\|$ *when running SYMMLQ and CG on two SPD problems for using various values of* $\lambda_{\text{est}} = \mu\lambda_{|\min|}$.

We observe in Figures 5a and 5c that for SYMMLQ, $\epsilon_k^L(\mu)/\|x_\star - x_k^L\| \approx \mu^{-1}$ after an initial lag. In the case of Bindel/ted_B_unscaled, an instability occurs for $\mu = 1 - 10^{-10}$ because the smallest eigenvalue is $\lambda_{|\min|} \approx 10^{-11}$. The instability is remedied by using a slightly larger $\mu = 1 - 10^{-4}$, which results in an almost identical bound, but without the instability.

For CG in Figures 5b and 5d, we also notice that for $\mu \leq 0.1$, the bound loosens by a factor of μ but keeps the same shape. The exception is when $\mu \approx 1$, where the bound is fairly tight until a divergence occurs and the bound nearly overlaps with the curve for $\mu = 0.1$. The closer μ is to 1, the later this divergence occurs; however, when $\lambda_{|\min|}$ is very small (as in Figure 5d), this may result in numerically unstable computations. This is because we are implicitly solving against the shifted system $T_k - \lambda_{\text{est}} I$ to compute the bound, which becomes singular as λ_{est} approaches $\lambda_{|min|}$. Meurant and Tichý (2015) observed similar instabilities for CG A-norm error bounds when the true error approaches the square root of machine precision.

8.5. Indefinite *A*. We now consider indefinite examples PARSEC/Na5 and HB/lshp3025 ($n = 5822$ and 3025, $\kappa(A) \approx 10^3$ and 10^4). The former contains few negative eigenvalues, while for the latter, nearly half of its spectrum is negative. Figure 6a shows that with the negative eigenvalue, (12) is no longer a bound for all iterations and behaves only as an estimate which often dips below the true error. However, for many problems, such as for HB/lshp3025 in Figure 6b, we see that the



(a) PARSEC/Na5 (b) HB/lshp3025

Fig. 6. *$\epsilon_k^L(\mu)/\left\|x_\star - x_k^L\right\|$ for two indefinite systems. The Gauss–Radau approach no longer guarantees an upper bound but works in some problems. The delay continues to provide a lower bound.*

error estimate using $\lambda_{|\min|}$ remains an upper bound (until convergence) and tracks the true error to nearly an order of magnitude. Underestimation of $\lambda_{|\min|}$ loosens the bound but, in the case of both problems here, keeps (12) an upper bound to the true error, although this is again heuristic.

9. Finite-precision considerations and termination criteria. We must remember that the previous sections assumed exact arithmetic, including global preservation of orthogonality of the columns of V_k. The question arises whether ϵ_k^L (16) and ϵ_k^C (20) remain upper bounds in finite precision. A rounding-error analysis is needed, similar to that of Strakoš and Tichý (2002) for CG A-norm error lower bounds, but this remains for future work. The rigorous analysis of Golub and Strakǒs (1994) shows that Gauss–Radau quadrature may not yield upper bounds in finite precision, yet its use in finite-precision computation remains justified. In all of our numerical experiments with positive semidefinite A, we have observed that the computed ϵ_k^L and ϵ_k^C are indeed upper bounds on the errors in x_k^L and x_k^C until convergence. It may therefore be possible to derive the error bounds in this paper only using assumptions of local orthogonality in the CG and Lanczos algorithms.

For positive semidefinite A, we have seen in practice that if λ_{est} is close to λ_r, the error bounds are remarkably tight. Heuristically, we observe that when λ_{est} is loose, $|\lambda_r|/|\lambda_{\text{est}}| \approx \epsilon_k^L/\|x_\star - x_k^L\|$. It was shown in sections 8.2–8.3 that the error estimate is an upper bound until convergence, after which the true error may plateau but ϵ_k^C and ϵ_k^L continue to decrease. This property makes it possible to terminate the iterations as soon as ϵ_k^L or ϵ_k^C drops below a prescribed level.

For CG with positive semidefinite A, we have seen that ϵ_k^C is typically one or two orders of magnitude larger than the true error for reasonable choices of λ_{est}. Using the ϵ_k^C termination criterion will ensure that the error satisfies some tolerance, but CG may take a few more iterations than necessary to achieve that tolerance.

For SYMMLQ with indefinite A, although ϵ_k^L is not guaranteed to upper bound the error, it still acts as a useful estimate of the error. Since ϵ_k^L may diverge from the exact values, if one monitors the residual, it will not be difficult to tell if ϵ_k^L is erroneously approaching zero. Since ϵ_k^L tends to upper bound the error near convergence, it can still be used in conjunction with other termination criteria involving the residual and related quantities to obtain solutions that probably satisfy a given error tolerance.

TABLE 2
Comparison of CG and SYMMLQ properties on a positive semidefinite consistent system $Ax = b$. Italicized results hold for indefinite systems as well.

	CG	SYMMLQ
$\|x_k\|$	$\nearrow$ (S, 1983, Theorem 2.1)	$\nearrow$ *(PS,* 1975*)*, $\leq$ CG (Theorem 6)
$\|x_\star - x_k\|$	$\searrow$ (HS, 1952, Theorem 6:3)	$\searrow$ *(PS,* 1975*)*, $\geq$ CG (Theorem 6)
$\|x_\star - x_k\|_A$	$\searrow$ (HS, 1952, Theorem 4:3)	not-monotonic
$\|r_k\|$	not-monotonic	not-monotonic
$\|r_k\| \,/\, \|x_k\|$	not-monotonic	not-monotonic

$\nearrow$ monotonically increasing $\searrow$ monotonically decreasing
S (Steihaug, 1983), HS (Hestenes and Stiefel, 1952), PS (Paige and Saunders, 1975)

10. Concluding remarks. We have developed cheap estimates for the error in SYMMLQ and CG iterates and explored the relationship between those errors. The main results are in (10)–(12), (15), and (20). The complete algorithm is summarized in Algorithm 1. Fong and Saunders (2012, Table 5.1) summarize the monotonicity of various quantities related to the CG and MINRES iterations. Table 2 is similar but focuses on CG and SYMMLQ.

When A is positive semidefinite, our error estimates are upper bounds prior to convergence (under exact arithmetic). For CG, the estimate can be made tighter by utilizing a delay d as described in (21), for an additional $O(d)$ flops and storage. When A is indefinite, the SYMMLQ estimate is not guaranteed to be an upper bound, but often tracks the error closely after an initial lag.

Acknowledgments. We are deeply grateful to the referees and associate editor for their insight and extremely helpful recommendations.

REFERENCES

M. Arioli (2013), *Generalized Golub–Kahan bidiagonalization and stopping criteria*, SIAM J. Matrix Anal. Appl., 34, pp. 571–592, https://doi.org/10.1137/120866543.

G. Auchmuty (1992), *A posteriori error estimates for linear equations*, Numer. Math., 61, pp. 1–6, https://doi.org/10.1007/BF01385494.

C. Brezinski (1999), *Error estimates for the solution of linear systems*, SIAM J. Sci. Comput., 21, pp. 764–781, https://doi.org/10.1137/S1064827597328510.

G. Dahlquist, S. C. Eisenstat, and G. H. Golub (1972), *Bounds for the error of linear systems of equations using the theory of moments*, J. Math. Anal. Appl., 37, pp. 151–166, https://doi.org/10.1016/0022-247X(72)90264-8.

G. Dahlquist, G. H. Golub, and S. G. Nash (1979), *Bounds for the error in linear systems*, in Semi-Infinite Programming, Lecture Notes in Control and Inform. Sci. 15, Springer, Berlin, New York, pp. 154–172.

T. A. Davis and Y. Hu (2011), *The University of Florida sparse matrix collection*, ACM Trans. Math. Software, 38, 1, https://doi.org/10.1145/2049662.2049663.

R. Estrin, D. Orban, and M. A. Saunders (2016), *LSLQ: An Iterative Method for Linear Least-Squares Problems with a Forward Error Minimization Property*, Cahier du GERAD G-2017-05, GERAD, Montréal, QC, Canada.

B. Fischer (1996), *Polynomial Based Iteration Methods for Symmetric Linear Systems*, Wiley–Teubner Ser. Adv. Numer. Math., John Wiley & Sons, Ltd., Chichester, UK, B. G. Teubner, Stuttgart, Germany, https://doi.org/10.1007/978-3-663-11108-5.

D. C.-L. Fong and M. A. Saunders (2012), *CG versus MINRES: An empirical comparison*, SQU J. Sci., 17, pp. 44–62.

A. Frommer, K. Kahl, Th. Lippert, and H. Rittich (2013), 2*-norm error bounds and estimates for Lanczos approximations to linear systems and rational matrix functions*, SIAM J. Matrix Anal. Appl., 34, pp. 1046–1065, https://doi.org/10.1137/110859749.

G. H. Golub and G. Meurant (1994), *Matrices, moments and quadrature*, in Numerical Analysis 1993 (Dundee, 1993), Pitman Res. Notes Math. Ser. 303, Longman Sci. Tech., Harlow, UK, pp. 105–156.

G. H. Golub and G. Meurant (1997), *Matrices, moments and quadrature* II; *how to compute the norm of the error in iterative methods*, BIT, 37, pp. 687–705, https://doi.org/10.1007/BF02510247.

G. H. Golub and G. Meurant (2009), *Matrices, Moments and Quadrature with Applications*, Princeton University Press, Princeton, NJ.

G. H. Golub and Z. Strakŏs (1994), *Estimates in quadratic formulas*, Numer. Algorithms, 8, pp. 241–268, https://doi.org/10.1007/BF02142693.

G. H. Golub and C. F. Van Loan (2013), *Matrix Computations*, 4th ed., Johns Hopkins University Press, Baltimore, MD.

M. R. Hestenes and E. Stiefel (1952), *Methods of conjugate gradients for solving linear systems*, J. Res. Natl. Bur. Stand., 49, pp. 409–436.

C. Lanczos (1950), *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Natl. Bur. Stand., 45, pp. 255–282.

G. Meurant (1997), *The computation of bounds for the norm of the error in the conjugate gradient algorithm*, Numer. Algorithms, 16, pp. 77–87, https://doi.org/10.1023/A:1019178811767.

G. Meurant (2005), *Estimates of the l_2 norm of the error in the conjugate gradient algorithm*, Numer. Algorithms, 40, pp. 157–169, https://doi.org/10.1007/s11075-005-1528-0.

G. Meurant (2006), *The Lanczos and Conjugate Gradient Algorithms: From Theory to Finite Precision Computations*, Software Environ. Tools 19, SIAM, Philadelphia, https://doi.org/10.1137/1.9780898718140.

G. Meurant and P. Tichý (2015), *On the Numerical Behavior of Quadrature Based Bounds for the A-Norm of the Error in CG*, http://www.cs.cas.cz/~tichy/download/present/2015ALA.pdf.

D. Orban and M. Arioli (2017), *Iterative Solution of Symmetric Quasi-Definite Linear Systems*, SIAM Spotlights 3, SIAM, Philadelphia, https://doi.org/10.1137/1.9781611974737.

C. C. Paige and M. A. Saunders (1975), *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12, pp. 617–629, https://doi.org/10.1137/0712047.

M. A. Saunders (1995), *Solution of sparse rectangular systems using LSQR and Craig*, BIT, 35, pp. 588–604, https://doi.org/10.1007/BF01739829.

M. A. Saunders (2016), *CME* 338 *class notes* 4: *Iterative methods for symmmetric $Ax = b$*, http://stanford.edu/class/msande318/notes.html.

T. Steihaug (1983), *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20, pp. 626–637, https://doi.org/10.1137/0720042.

Z. Strakoš and P. Tichý (2002), *On error estimation in the conjugate gradient method and why it works in finite precision computations*, Electron. Trans. Numer. Anal., 13, pp. 56–80.

D. B. Szyld and O. B. Widlund (1993), *Variational analysis of some conjugate gradient methods*, East-West J. Numer. Math., 1, pp. 51–74.