SIAM J. MATRIX ANAL. APPL.
Vol. 41, No. 3, pp. 1183–1207

SHARP 2-NORM ERROR BOUNDS FOR LSQR AND THE CONJUGATE GRADIENT METHOD*

ERIC HALLMAN†

Abstract. We consider the iterative method LSQR for solving $\min_x \|Ax-b\|_2$. LSQR is based on the Golub–Kahan bidiagonalization process and at every step produces an iterate that minimizes the norm of the residual vector over a Krylov subspace $\mathcal{K}_k$. The 2-norm of the error is known to decrease monotonically, although it is not minimized over $\mathcal{K}_k$. Given a lower bound on the smallest singular value of A, we show that in exact arithmetic the solution lies in the interior of a certain ellipsoid and that the LSQR iterate lies on the boundary of this ellipsoid. We use this result to derive new 2-norm error bounds for LSQR. Although our bounds are not much smaller than the existing ones, we show that they are sharp in the following sense: if the only information we use is our lower bound on $\sigma_{\min}(A)$ plus the information gained by running k steps of LSQR, then our bounds cannot be improved. We also show how to choose a point with an error bound smaller than our corresponding bound for the LSQR error, although its true error is not necessarily smaller than the true LSQR error. As LSQR is formally equivalent to the conjugate gradient (CG) method applied to the normal equations $A^TAx = A^Tb$, we derive analogous error bounds for CG. Our bounds for CG apply to any system $Ax = b$ where A is symmetric positive definite.

Key words. LSQR, least-squares problem, sparse matrix, Krylov subspace method, Golub–Kahan process, conjugate gradient method, stopping criteria, iterative method

AMS subject classifications. 15A06, 65F10, 65F20, 65F50, 93E24

DOI. 10.1137/19M1272822

1. Introduction.

We examine LSQR [21], an iterative method for solving the least-squares problem

$$\min_x \|Ax-b\|_2 \tag{LS}$$

for a matrix $A \in \mathbb{R}^{m\times n}$ and vector $b \in \mathbb{R}^m$. We require A only to compute matrix-vector products of the form Av and A^Tu, and these matrix-vector products typically dominate the cost of the algorithm.

LSQR is based on a bidiagonalization process by Golub and Kahan [8, eq. (2.4)].[1] Assuming $x_0^Q = 0$, LSQR produces an iterate x_k^Q at the kth step that minimizes the residual norm $\|r_k^Q\|_2 := \|b - Ax_k^Q\|_2$ over the Krylov subspace

$$\mathcal{K}_k\left(A^TA, A^Tb\right) = \text{Span}\left\{A^Tb, \left(A^TA\right)A^Tb, \ldots, \left(A^TA\right)^{k-1}A^Tb\right\}.$$

In exact arithmetic, it will terminate in at most $\min\{m,n\}$ steps and return

$$x_* := A^\dagger b,$$

*Received by the editors July 5, 2019; accepted for publication (in revised form) by D. Orban June 25, 2020; published electronically August 11, 2020.
https://doi.org/10.1137/19M1272822
Funding: The work of the author was partly supported by National Science Foundation grant DMS-1745654.

†Department of Mathematics, North Carolina State University, Raleigh, NC 27607 (erhallma@ncsu.edu).

[1]The paper by Golub and Kahan introduces multiple methods for reducing a matrix A to a bidiagonal matrix. The one we refer to is an iterative procedure often called the Golub–Kahan–Lanczos method, which we will call the *Golub–Kahan process*.

the minimum-norm solution to the least-squares problem. This will often fail to happen in practice due to roundoff errors, so we are interested in monitoring the error $\|x_k^Q - x_*\|_2$ in order to know when the algorithm may be safely halted.

In exact arithmetic, LSQR is equivalent to running the conjugate gradient (CG) method [12] on the normal equations $A^TAx = A^Tb$. We therefore simultaneously consider using CG to solve the problem

$$\bar{A}x = \bar{b}, \tag{SPD}$$

where $\bar{A}$ is an arbitrary symmetric positive definite (SPD) matrix. As was the case with previous work in bounding the error for LSQR (resp., CG) [4, 9, 14, 15, 7], we require a nontrivial lower bound $\tilde{\sigma} \le \sigma_{\min}(A)$ (resp., $\tilde{\sigma} \le \sqrt{\lambda_{\min}(\bar{A})}$).

It is known that the iterates x_k^Q for LSQR (and CG) are updated along positively correlated directions (i.e., if $x_k^Q = x_{k-1}^Q + p_k$, then $p_i^T p_j > 0$ for all i and j) and therefore that the 2-norm of the error decreases monotonically [12, Thms. 5:3, 6:3], although it is not minimized over $\mathcal{K}_k(A^TA, A^Tb)$. Recently, Estrin, Orban, and Saunders [4] developed an error estimate that took advantage of these properties. Building on work by Golub and Meurant [9] and Meurant [14, 15], they showed how to use $\tilde{\sigma}$ to cheaply compute an upper bound on $\|x_*\|_2$ and used it to derive an upper bound on the LSQR error [4, sect. 4.2]. In the same paper they developed the algorithm LSLQ as an auxiliary to LSQR, allowing their error estimate to be computed more stably. As LSLQ is equivalent to SYMMLQ [19] run on the normal equations, this paper paralleled their earlier work in using SYMMLQ to estimate the CG error [2].

Work by Meurant, Tichý, and others (see [17, eq. (6)] and the references there) used $\tilde{\sigma}$ to compute an upper bound for the A-norm of the CG error. This bound may be used to derive the 2-norm error bound

$$\left\|x_k^{CG} - x_*\right\|_2 \le \frac{1}{\tilde{\sigma}} \left\|x_k^{CG} - x_*\right\|_{\bar{A}} < \frac{|\tilde{\phi}_{k+1}|}{\tilde{\sigma}}, \tag{1.1}$$

where $\tilde{\phi}_{k+1}$ is a cheaply computable quantity depending on $\tilde{\sigma}$.

1.1. Summary of main results. In this paper we improve the existing error bounds for LSQR and CG. We start by identifying at each step k a point $\tilde{x}_{k+1}$, whose value depends on $\tilde{\sigma}$, such that the bound given by Estrin, Orban, and Saunders [4, Thm. 4] may be written (Theorem 3.1) in the form

$$\|x_*\|_2 \le \|\tilde{x}_{k+1}\|_2.$$

Although we cannot exactly find the error-minimizing points

$$x_k^* := \underset{x \in \mathcal{K}_k(A^TA, A^Tb)}{\arg\min} \|x - x_*\|_2, \tag{1.2}$$

we show (Corollary 3.3) that x_{k+1}^* is a convex combination of x_k^Q and $\tilde{x}_{k+1}$. Furthermore, we show (Theorem 3.4) that the solution $x_* = A^\dagger b$ lies in an ellipsoid centered at the point

$$\tilde{x}_{k+1}^{\mathcal{E}} := \frac{1}{2}\left(x_k^Q + \tilde{x}_{k+1}\right).$$

In the least-squares case, we also show that x_{k+1}^* is a convex combination of x_k^Q and the iterate x_{k+1}^G from a related algorithm known as *Craig's method* [5, 18, 21]. These bounds are illustrated in Figures 3.1 and 3.2.

We use these results to derive a new error bound for general points in the Krylov subspace (equation (3.24)) and in particular new error bounds for LSQR and CG (section 3.5). We also find (equation (3.25)) that the point $\tilde{x}^{\mathcal{E}}_{k+1}$ yields the error bound

$$\left\|\tilde{x}^{\mathcal{E}}_{k+1} - x_*\right\|_2 \leq \frac{|\tilde{\phi}_{k+1}|}{2\tilde{\sigma}},$$

which is precisely a factor of 2 smaller than the bound in (1.1).

More importantly, we demonstrate (Theorem 3.8) that our bounds are sharp. Specifically, we are interested in error bounds that satisfy two properties:

- They rely only on $\tilde{\sigma}$ and the information gained from running LSQR or CG.
- In exact arithmetic, they are provably upper bounds on the true error.

Of all such bounds, ours are the tightest possible. This result implies that future error estimates should either use additional information about A and b or settle for being estimates rather than guaranteed upper bounds.

We also assume exact arithmetic throughout. This approach limits the scope of our paper, since the bidiagonalization process used by LSQR produces a sequence of vectors that are orthogonal in exact arithmetic but may quickly lose that orthogonality in practice. It also stands in contrast to prior works such as [17] which have derived estimates that rely only on the local orthogonality of these vectors rather than global orthogonality. Although numerical experiments suggest that our bounds are reliable in practice, a more thorough exploration of how they behave in finite precision will be left to future research.

1.2. Organization. Section 2 summarizes the Golub–Kahan process and some of the relations between LSQR, CG, and Craig's method. Section 3 introduces our new error bound, proves that it is sharp, and shows how to choose the point that will minimize our error estimate. Section 4 discusses some limitations that our bounds face in practice, and section 5 summarizes our main results in notation more common to CG. Section 6 discusses regularization. Section 7 shows the results of some numerical experiments, and section 8 offers our concluding remarks.

1.3. Notation. We use Householder notation in general, denoting matrices, vectors, and scalars by A, a, and α, respectively. One exception is in describing Givens rotations, where c and s are used to denote the significant components of the rotation. The vector e_k always refers to the kth column of the identity matrix I_d, where d can be inferred from context. Writing the compact SVD of A as $U\Sigma V^T$, we denote the projection onto the column space of A by $\Pi_A = UU^T$ and the pseudoinverse of A by $A^\dagger = V\Sigma^{-1}U^T$. The smallest nonzero singular value of A is denoted by $\sigma_{\min}(A)$. If M is a positive definite matrix, then $\|v\|_M = \sqrt{v^T M v}$. The notation $A \succeq B$ means that $A - B$ is positive semidefinite.

In this paper we discuss several closely related algorithms and use several QR and QL factorizations. We use superscripts to distinguish the various algorithms, in particular using x^Q_k (and y^Q_k, r^Q_k, etc.) for quantities related to LSQR, x^G_k for Craig's method, and x^*_k for the iterate minimizing the 2-norm error over $\mathcal{K}_k$. For a more comprehensive list, see Figure 1.1. Typically, by "the error" of an iterate x_k we mean the quantity $\|x_k - x_*\|_2$.

Diacritics ($\ddot{Q}$, $\hat{R}$) generally distinguish the QR (or QL) factorizations as well as associated scalars ($\ddot{\theta}$, $\hat{\rho}$). The leading elements of the R factors do not change from one iteration to the next, and transient elements will be denoted using prime notation.

x_k^Q	The kth LSQR iterate.
r_k^Q	The LSQR residual $b - Ax_k^Q$.
V_k	Spans the kth Krylov subspace: e.g., $x_k^Q = V_k y_k^Q$.
h_{k+1}	Update direction from x_k^Q to x_{k+1}^Q, x_{k+1}^G, x_{k+1}^*, $\tilde{x}_{k+1}$, and $\tilde{x}_{k+1}^{\mathcal{E}}$.
x_k^G	The kth iterate from Craig's method.
x_*	The minimum-norm solution to LS. Defined by $x_* = A^\dagger b$.
x_{k+1}^*	The projection of x_* onto the column space of V_{k+1}.
$\mathcal{E}_{k+1}$	An ellipsoid that contains x_*. Used to derive error bounds for (SPD).
$\mathcal{E}_{k+1}^{(G)}$	Intersection of $\mathcal{E}_{k+1}$ and a half-space related to Craig's method. Used to derive error bounds for (LS).
$\tilde{x}_{k+1}$	Lies on the boundary of $\mathcal{E}_{k+1}$, opposite x_k^Q. Satisfies $\|x_*\|_2 \le \|\tilde{x}_{k+1}\|_2$.
$\tilde{x}_{k+1}^{\mathcal{E}}$	The center of $\mathcal{E}_{k+1}$. Minimizes our error bound for (SPD).
$\tilde{x}_{k+1}^{\mathcal{E}(G)}$	Either $\tilde{x}_{k+1}^{\mathcal{E}}$ or x_{k+1}^G, whichever has smaller norm. Minimizes our error bound for (LS).
$\tilde{\sigma}$	A lower bound for $\sigma_{\min}(A)$.
ϕ_{k+1}^*	Defined so that $\|\Pi_A r_k^Q\|_2 = \|\phi_{k+1}^*\|$. Related to the location of x_{k+1}^*.
ϕ'_{k+1}	In exact arithmetic, satisfies $\|r_k^Q\|_2 = \|\phi'_{k+1}\|$.
$\tilde{\phi}_{k+1}$	In exact arithmetic, satisfies $\|\Pi_A r_k^Q\|_2 \le \|\tilde{\phi}_{k+1}\|$.
$\tilde{\rho}_{k+1}$	Chosen so that $\sigma_{\min}(\widetilde{R}_{k+1}) = \tilde{\sigma}$. Used to compute $\tilde{x}_{k+1}$ and $\tilde{x}_{k+1}^{\mathcal{E}}$.
$\tilde{c}_{k+1}$	Equal to $\tilde{\phi}_{k+1}/\phi'_{k+1}$. If $\tilde{c}_{k+1} < 1$, the system must be inconsistent.

FIG. 1.1. *Summary of notation appearing in this paper.*

Such elements $(\phi'_k,\ \rho'_{k+1})$ will typically change into a related element $(\phi_k,\ \rho_{k+1})$ in the next iteration. Quantities such as $\widetilde{R}, \tilde{x}, \tilde{\rho}, \tilde{\phi}$ all depend on the lower bound $\tilde{\sigma}$, and quantities such as $\bar{A}, \bar{b}, \bar{r}$ refer to (SPD) as opposed to (LS).

2. Background. We start with a short summary of the Golub–Kahan bidiagonalization process [8, eq. (2.4)], an iterative method originally designed to estimate the singular values of a matrix A by reducing it to a lower bidiagonal matrix B. We use this process to reproduce the derivations of LSQR and Craig's method, and to introduce notation that will be used in the remainder of the paper.

2.1. The Golub–Kahan process. The Golub–Kahan process takes a matrix A and vector b and after k steps produces orthogonal matrices $U_k = [u_1, \ldots, u_k]$ and $V_k = [v_1, \ldots, v_k]$ such that

$$\begin{aligned}
\operatorname{Span}(U_k) &= \operatorname{Span}\left\{b, \left(AA^T\right) b, \ldots, \left(AA^T\right)^{k-1} b\right\} = \mathcal{K}_k\left(AA^T, b\right), \\
\operatorname{Span}(V_k) &= \operatorname{Span}\left\{A^T b, \left(A^T A\right) A^T b, \ldots, \left(A^T A\right)^{k-1} A^T b\right\} = \mathcal{K}_k\left(A^T A, A^T b\right).
\end{aligned}$$

The process itself proceeds according to Algorithm 2.1. In the event that $\alpha_{k+1} = 0$ or $\beta_{k+1} = 0$, the process terminates and we can solve the least-squares problem exactly. The case $\beta_{k+1} = 0$ additionally implies that the system $Ax = b$ is consistent.

Algorithm 2.1 Golub–Kahan bidiagonalization process

Require: A, b

$\beta_1 u_1 = b$ ▷ $\beta_1 = \|b\|$, $u_1 = b/\beta_1$

$\alpha_1 v_1 = A^T u_1$ ▷ $\alpha_1 = \|A^T u_1\|$, $v_1 = A^T u_1/\alpha_1$

for $k = 1, 2, \ldots$ **do**

  $\beta_{k+1} u_{k+1} = A v_k - \alpha_k u_k$

  $\alpha_{k+1} v_{k+1} = A^T u_{k+1} - \beta_{k+1} v_k$

end for

By defining the lower bidiagonal matrices

$$L_k = \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \end{bmatrix}, \quad B_k = \begin{bmatrix} L_k \\ \beta_{k+1} e_k^T \end{bmatrix},$$

we can characterize the process at each iteration by the two relations

$$AV_k = U_{k+1} B_k \quad \text{and} \quad A^T U_{k+1} = V_{k+1} L_{k+1}^T, \tag{2.1}$$

which remain accurate even in finite precision.

2.2. Subproblem for LSQR. At every iteration, each algorithm in this paper produces an iterate from the space $\mathcal{K}_k(A^T A, A^T b) = \mathrm{Span}(V_k)$, and so (with $x_0 = 0$) their iterates may be written in the form $x_k = V_k y_k$ for some $y_k \in \mathbb{R}^k$. For LSQR, the iterate x_k^Q is chosen to minimize the residual norm $\|r_k^Q\|_2$ at every step. Using the first relation in (2.1), we find that for any $x_k \in \mathrm{Span}(V_k)$ we have

$$r_k = b - Ax_k = b - AV_k y_k = \beta_1 u_1 - U_{k+1} B_k y_k = U_{k+1}(\beta_1 e_1 - B_k y_k),$$

and so assuming the orthogonality of U_{k+1} (which holds in exact arithmetic), it follows that

$$\min_{x_k = V_k y_k} \|r_k\|_2 = \min_{y_k} \|B_k y_k - \beta_1 e_1\|_2 . \tag{2.2}$$

Since B_k is a lower bidiagonal matrix, we can solve this problem efficiently by finding its QR factorization.

2.3. QR factorization. The Q factor from the QR factorization of B_k can be expressed as the product

$$Q_k = P_k \ldots P_2 P_1,$$

where P_i is a plane rotation acting on rows i and $i+1$ of a given matrix and having significant component $\left[\begin{smallmatrix} c_i & s_i \\ -s_i & c_i \end{smallmatrix}\right]$.[2] The QR factorization then takes the form

$$Q_k \begin{bmatrix} B_k & \beta_1 e_1 \end{bmatrix} = \begin{bmatrix} R_k & f_k \\ 0 & \phi'_{k+1} \end{bmatrix} = \left[\begin{array}{cccc:c} \rho_1 & \theta_2 & & & \phi_1 \\ & \rho_2 & \ddots & & \phi_2 \\ & & \ddots & \theta_k & \vdots \\ & & & \rho_k & \phi_k \\ \hline & & & & \phi'_{k+1} \end{array}\right], \tag{2.3}$$

[2] This involves an abuse of notation since at the kth step each matrix P_i will be $(k+1) \times (k+1)$; thus P_i does not have a single fixed size. The significant component of P_i, however, does not change.

where the plane rotation P_k has the effect

$$\begin{bmatrix} c_k & s_k \\ -s_k & c_k \end{bmatrix} \begin{bmatrix} \rho'_k & 0 & \phi'_k \\ \beta_{k+1} & \alpha_{k+1} & 0 \end{bmatrix} = \begin{bmatrix} \rho_k & \theta_{k+1} & \phi_k \\ 0 & \rho'_{k+1} & \phi'_{k+1} \end{bmatrix} \tag{2.4}$$

with $\rho'_1 = \alpha_1$ and $\phi'_1 = \beta_1$. We will additionally use the notation

$$Q_k L_{k+1} = R'_{k+1} = \begin{bmatrix} R_k & \theta_{k+1} e_k \\ 0 & \rho'_{k+1} \end{bmatrix}, \tag{2.5}$$

$$Q_k(\beta_1 e_1) = f'_{k+1} = \begin{bmatrix} f_k \\ \phi'_{k+1} \end{bmatrix} \tag{2.6}$$

so that R'_{k+1} and f'_{k+1} are identical to R_{k+1} and f_{k+1} except in the final element.

2.4. Solution for LSQR. To solve the LSQR subproblem (2.2), we use the QR factorization from (2.3) to find that for all $x_k \in \mathrm{Span}(V_k)$,

$$\|r_k\|_2 = \|B_k y_k - \beta_1 e_1\|_2 = \left\| \begin{bmatrix} R_k \\ 0 \end{bmatrix} y_k - \begin{bmatrix} f_k \\ \phi'_{k+1} \end{bmatrix} \right\|_2.$$

LSQR minimizes this expression by solving $R_k y_k^Q = f_k$ and gives the residual norm $\|r_k^Q\|_2 = |\phi'_{k+1}|$, although we do not need to compute either y_k^Q or r_k^Q explicitly.

If we solve $R_k^T W_k^T = V_k^T$ by forward substitution and define w_k to be the last column of W_k, we get the recurrence relation

$$x_k^Q = V_k y_k^Q = V_k R_k^{-1} f_k = W_k f_k = x_{k-1}^Q + \phi_k w_k.$$

As mentioned in [21, sect. 4.1], it is computationally more efficient to define $h_k = \rho_k w_k$ and use the recurrence

$$x_k^Q = x_{k-1}^Q + \frac{\phi_k}{\rho_k} h_k. \tag{2.7}$$

It turns out that using h_k also makes our formulas slightly cleaner, so we will use h_k instead of w_k for the remainder of the paper. The vector h_k can be computed by the recurrence

$$\frac{\theta_{k+1}}{\rho_k} h_k + h_{k+1} = v_{k+1},$$

where $h_1 = v_1$. Algorithm 2.2 provides pseudocode for LSQR, with only minor changes from its presentation in [21].

2.5. Craig's method. Craig's method [21, sect. 7.2] solves the consistent system $Ax = b$ by using forward substitution to solve the problem

$$L_k y_k^G = \beta_1 e_1. \tag{2.8}$$

The leading coordinates of y_k^G do not change, and so $x_k^G = V_k y_k^G$ updates along orthogonal directions. This implies that Craig's method minimizes the error $\|x_k - x_*\|_2$ at each step, provided the system is consistent.

Craig's method and LSQR are respectively equivalent [3] to running CG and MINRES on the normal equations

$$AA^T y = b,\ x = A^T y,$$

Algorithm 2.2 LSQR

Require: A, b

$\beta_1 u_1 = b$, $\alpha_1 v_1 = A^T u_1$
$\phi'_1 = \beta_1$, $\rho_0 = 1$, $\rho'_1 = \alpha_1$
$x_0^Q = 0$, $h_1 = v_1$
for $k = 1, 2, \ldots$ **do**
  $\beta_{k+1} u_{k+1} = A v_k - \alpha_k u_k$ ▷ Continue the bidiagonalization
  $\alpha_{k+1} v_{k+1} = A^T u_{k+1} - \beta_{k+1} v_k$
  $\rho_k = \left(\rho_k'^2 + \beta_{k+1}^2\right)^{1/2}$ ▷ Construct and apply rotation P_k
  $c_k = \rho'_k / \rho_k$, $s_k = \beta_{k+1} / \rho_k$
  $\theta_{k+1} = s_k \alpha_{k+1}$, $\rho'_{k+1} = c_k \alpha_{k+1}$
  $\phi_k = c_k \phi'_k$, $\phi'_{k+1} = -s_k \phi'_k$
  $x_k^Q = x_{k-1}^Q + (\phi_k / \rho_k) h_k$ ▷ Update h and x_k^Q
  $h_{k+1} = v_{k+1} - (\theta_{k+1} / \rho_k) h_k$
end for

and as with CG and MINRES it is possible to cheaply transfer between the two. For our purposes, it is simplest to consider the transfer from x_k^Q to x_{k+1}^G. Extending (2.8) one iterate further and performing the QR factorization (2.5) implies that

$$\begin{bmatrix} R_k & \theta_{k+1} e_k \\ 0 & \rho'_{k+1} \end{bmatrix} y_{k+1}^G = \begin{bmatrix} f_k \\ \phi'_{k+1} \end{bmatrix}$$

and therefore, using the notation from (2.5) and (2.6), that

$$x_{k+1}^G = V_{k+1} R_{k+1}'^{-1} f'_{k+1}.$$

It follows that

$$x_{k+1}^G = V_{k+1} R_{k+1}^{-1} \begin{bmatrix} f_k \\ \rho_{k+1} \phi'_{k+1} / \rho'_{k+1} \end{bmatrix} = x_k^Q + \frac{\phi'_{k+1}}{\rho'_{k+1}} h_{k+1}. \tag{2.9}$$

By (2.4), ϕ_{k+1} and ϕ'_{k+1} have the same sign. Comparing this transfer with the update for x_{k+1}^Q from (2.7), we conclude that x_k^Q, x_{k+1}^Q, and x_{k+1}^G are collinear and appear in that order.

Figure 2.1 illustrates the basic geometric relations between the LSQR and Craig iterates, along with the LSLQ iterates x_k^L (discussed further in [4, 10]) and the optimal points x_k^* (discussed in section 3).

2.5.1. Craig's method for least-squares problems. An early method for extending Craig's method to least-squares problems was introduced by Paige in [18] and discussed by Paige and Saunders [20, sect. 7.3], where the authors found that the method was equivalent to transferring to the LSQR point. As transferring in this direction was highly unstable on inconsistent problems, the authors recommended discarding the method.

Saunders [22] later proposed a method called extended CRAIG to solve the regularized least-squares problem

$$\min_x \left\| \begin{bmatrix} A \\ \lambda I \end{bmatrix} x - \begin{bmatrix} b \\ 0 \end{bmatrix} \right\|_2$$

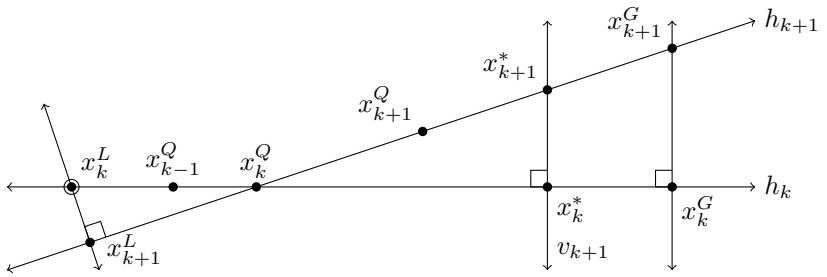

FIG. 2.1. *The two-dimensional affine space* $x_k^L + \mathrm{Span}\{h_k, v_{k+1}\}$. *The LSLQ point* x_k^L *has minimal norm within this space. Distances are not to scale, but all relative positions are correct except possibly for* 180*-degree rotations of the directions for* h_k, h_{k+1}, *or* v_{k+1}.

by using Craig's method on the equivalent[3] problem

$$\min_{x,s} \|x\|_2^2 + \|s\|_2^2 \quad \text{subject to} \quad \begin{bmatrix} A & \lambda I \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} = b.$$

However, this method minimizes $\|x_k - x_*\|_2^2 + \|s_k - s_*\|_2^2$ at each step rather than $\|x_k - x_*\|_2$, and in fact $\|x_k - x_*\|_2$ is not even necessarily monotonic.

2.6. The conjugate gradient method. The CG method [12, 2] was originally designed to solve the problem (SPD) where $\bar{A}$ is a positive definite matrix. At the kth iteration, the iterate x_k^{CG} lies in the Krylov subspace

$$\mathcal{K}_k(\bar{A}, \bar{b}) = \mathrm{Span}\left\{\bar{b}, \bar{A}\bar{b}, \ldots, \bar{A}^{k-1}\bar{b}\right\}.$$

The method can be derived from the Lanczos process [13], which is characterized by the relations

$$\bar{A}V_k = V_k T_k + \bar{\beta}_{k+1} v_{k+1} e_k^T = V_{k+1} H_k,$$

where $V_k = [v_1, \ldots, v_k]$ is orthogonal in exact arithmetic and

$$T_k = \begin{bmatrix} \bar{\alpha}_1 & \bar{\beta}_2 & & \\ \bar{\beta}_2 & \bar{\alpha}_2 & \ddots & \\ & \ddots & \ddots & \bar{\beta}_k \\ & & \bar{\beta}_k & \bar{\alpha}_k \end{bmatrix}.$$

In this case, we define $x_k^{CG} = V_k y_k^{CG}$, where $T_k y_k^{CG} = \bar{\beta}_1 e_1$ and $\bar{\beta}_1 = \|\bar{b}\|_2$. If $\bar{A} = A^T A$ and $\bar{b} = A^T b$, then in exact arithmetic LSQR and CG produce the same V_k, and $x_k^Q = x_k^{CG}$. To connect the two sets of notation further, we note that

$$\begin{aligned} T_k &= B_k^T B_k = R_k^T R_k, \\ H_k &= L_{k+1}^T B_k = \begin{bmatrix} R_k^T \\ \theta_{k+1} e_k^T \end{bmatrix} R_k, \\ f_k &= R_k^{-T}(\bar{\beta}_1 e_1) = R_k^{-T}(\alpha_1 \beta_1 e_1). \end{aligned}$$

In particular,

$$x_k^{CG} = V_k T_k^{-1}(\bar{\beta}_1 e_1) = V_k R_k^{-1} R_k^{-T}(\bar{\beta}_1 e_1) = V_k R_k^{-1} f_k.$$

[3] Via the substitution $s = \frac{1}{\lambda}(b - Ax)$.

There is at least one significant difference between the two processes. After k iterations the Golub–Kahan process gives enough information to compute the matrix R'_{k+1}, whose final element is ρ'_{k+1}. The Lanczos process produces H_k, which gives just enough information to compute R_k and θ_{k+1}. There therefore does not appear to be any clear analogue to Craig's method in the way that CG is analogous to LSQR, although we could express the Craig iterates in the more CG-like form

$$x^G_{k+1} = V_{k+1} T'^{-1}_{k+1}(\bar{\beta}_1 e_1), \quad \text{where} \quad T'_{k+1} = R'^T_{k+1} R'_{k+1}. \tag{2.10}$$

3. Bounding the error. The error bounds derived by Estrin, Orban, and Saunders in [4] rely primarily on finding an upper bound on $\|x_*\|_2$. The following theorem is a consequence of one of their main results [4, Thm. 4] as it pertains to this paper.

THEOREM 3.1 (see Estrin, Orban, and Saunders [4]). *Fix $\tilde{\sigma} \le \sigma_{\min}(A)$, and define*

$$\widetilde{R}_{k+1} := \begin{bmatrix} R_k & \theta_{k+1} e_k \\ 0 & \tilde{\rho}_{k+1} \end{bmatrix}, \tag{3.1}$$

where $\tilde{\rho}_{k+1} > 0$ is chosen so that $\sigma_{\min}(\widetilde{R}_{k+1}) = \tilde{\sigma}$. Also define $\tilde{\phi}_{k+1}$ so that

$$\widetilde{R}^T_{k+1} \tilde{f}_{k+1} := \begin{bmatrix} R_k & \theta_{k+1} e_k \\ 0 & \tilde{\rho}_{k+1} \end{bmatrix}^T \begin{bmatrix} f_k \\ \tilde{\phi}_{k+1} \end{bmatrix} = \alpha_1 \beta_1 e_1. \tag{3.2}$$

Then $\|x_*\|_2 \le \|\tilde{x}_{k+1}\|_2$, where

$$\tilde{x}_{k+1} := V_{k+1} \widetilde{R}^{-1}_{k+1} \tilde{f}_{k+1} = x^Q_k + \frac{\tilde{\phi}_{k+1}}{\tilde{\rho}_{k+1}} h_{k+1}. \tag{3.3}$$

A few comments are in order. First, it can be seen from (2.3) that

$$R^T_k f_k = B^T_k(\beta_1 e_1) = \alpha_1 \beta_1 e_1$$

and therefore that $\tilde{f}_{k+1}$ is well-defined. Second, $|\tilde{\phi}_{k+1}|$ is known to be an upper bound on $\|\Pi_A r^Q_k\|_2$ for (LS) and on $\|x^{CG}_k - x_*\|_{\bar{A}}$ for (SPD) [16, 17, 11], and this fact leads immediately to the error bound in (1.1). Finally, it was shown in [16, eq. (3.10)] that $\tilde{\rho}_{k+1}$ satisfies the recurrence

$$\tilde{\rho}^2_{k+1} = \tilde{\sigma}^2 + \frac{\theta^2_{k+1} \tilde{\rho}^2_k}{\rho^2_k - \tilde{\rho}^2_k}, \tag{3.4}$$

where $\tilde{\rho}_1 = \tilde{\sigma}$. The quantities $\tilde{\rho}_{k+1}$ and $\tilde{\phi}_{k+1}$ satisfy the relations

$$\tilde{\rho}_{k+1} \tilde{\phi}_{k+1} = \rho'_{k+1} \phi'_{k+1} = \rho_{k+1} \phi_{k+1}.$$

3.1. Decomposition lemma. With the aim of improving upon existing error bounds, we begin by strengthening the results of Theorem 3.1. Consider the orthogonal decomposition

$$x_* = x^*_{k+1} + x^\perp_{k+1},$$

where x^*_{k+1} (the error-minimizing point from (1.2)) is the projection of x_* onto $\mathrm{Span}(V_{k+1})$. Then the following lemma gives us expressions for both x^*_{k+1} and $x^\perp_{k+1}$ in terms of a few unknown parameters.

LEMMA 3.2. *Assume that $x_* \notin \mathrm{Span}(V_k)$, so that Algorithm 2.1 has not halted after the first k iterations. First, define ϕ^*_{k+1} and ρ^*_{k+1} so that $|\phi^*_{k+1}| = \|\Pi_A r^Q_k\|_2$, $\rho^*_{k+1} > 0$, and*

$$R^{*T}_{k+1} f^*_{k+1} := \begin{bmatrix} R_k & \theta_{k+1} e_k \\ 0 & \rho^*_{k+1} \end{bmatrix}^T \begin{bmatrix} f_k \\ \phi^*_{k+1} \end{bmatrix} = \alpha_1 \beta_1 e_1. \tag{3.5}$$

Then

$$x^*_{k+1} = V_{k+1} R^{*-1}_{k+1} f^*_{k+1} = x^Q_k + \frac{\phi^*_{k+1}}{\rho^*_{k+1}} h_{k+1}. \tag{3.6}$$

Second, let ξ and $v^\perp$ be the unique nonnegative scalar and unit vector[4] satisfying

$$x^\perp_{k+1} = \xi \frac{\phi^*_{k+1}}{\rho^*_{k+1}} v^\perp. \tag{3.7}$$

There then exists a nonnegative scalar $\ddot{\rho}_{k+2}$ so that the matrix

$$\ddot{R}_{k+2} := \begin{bmatrix} R_k & \theta_{k+1} e_k & 0 \\ 0 & \rho^*_{k+1} & 0 \\ 0 & \xi \ddot{\rho}_{k+2} & \ddot{\rho}_{k+2} \end{bmatrix} \tag{3.8}$$

satisfies the bound

$$\sigma_{\min}(\ddot{R}_{k+2}) \geq \sigma_{\min}(A). \tag{3.9}$$

Finally, in the least-squares case, ρ^*_{k+1} will also satisfy $\rho^*_{k+1} \geq \rho'_{k+1}$.

Proof. In exact arithmetic, the bidiagonalization process will eventually terminate (say, after $t > k+1$ iterations)[5] and produce a matrix V_t that spans an invariant subspace of $A^T A$. If $AV_t = U_{t+1} B_t$, then we may use a QR factorization to convert B_t to the upper bidiagonal matrix R_t and obtain the equation

$$\|r_t\|_2 = \left\| \begin{bmatrix} f_t - R_t y_t \\ \phi'_{t+1} \end{bmatrix} \right\|_2.$$

Thus $y_* = y^Q_t = R_t^{-1} f_t$. We then perform a series of Givens rotations so that

$$\dot{Q}_t f_t = \dot{P}_{k+1} \dot{P}_{k+2} \cdots \dot{P}_{t-2} \dot{P}_{t-1} f_t = \begin{bmatrix} f_k \\ \phi^*_{k+1} \\ 0 \end{bmatrix}, \tag{3.10}$$

where each $\dot{P}_i$ acts on rows i and $i+1$. Since ϕ'_{t+1} represents the part of the residual outside the span of A and since the LSQR iterate satisfies $f_k - R_k y^Q_k = 0$, it follows that

$$|\phi^*_{k+1}| = \|[\phi_{k+1}, \phi_{k+2}, \ldots, \phi_t]\|_2 = \|\Pi_A r^Q_k\|_2,$$

as we defined it. Thus the use of ϕ^*_{k+1} in (3.10) is justified. We may then rewrite the product $R^T_t f_t$ in the form

[4] If $x^\perp_{k+1} = 0$, we set $\xi = 0$ and $v^\perp = 0$.

[5] It is simple to check the case $t = k+1$ separately.

$$R_t^T f_t = (\dot{Q}_t R_t)^T (\dot{Q}_t f_t) = \begin{bmatrix} R_k^T & 0 & 0 \\ \theta_{k+1} e_k^T & \rho_{k+1}^* & \dot{\theta}_{k+2} e_1^T \\ 0 & z & \dot{L}_t^T \end{bmatrix} \begin{bmatrix} f_k \\ \phi_{k+1}^* \\ 0 \end{bmatrix}. \tag{3.11}$$

Now $R_t^T f_t = B_t^T(\beta_1 e_1) = \alpha_1\beta_1 e_1$, and so equating entries of the leftmost and rightmost expressions in (3.11) reveals that $z = 0$. The form of the entry $\dot{\theta}_{k+2} e_1^T$ is justified because $\dot{Q}_t R_t$ is necessarily upper Hessenberg. It follows that

$$y_* = R_t^{-1} f_t = \begin{bmatrix} R_k & \theta_{k+1} e_k & 0 \\ 0 & \rho_{k+1}^* & 0 \\ 0 & \dot{\theta}_{k+2} e_1 & \dot{L}_t \end{bmatrix}^{-1} \begin{bmatrix} f_k \\ \phi_{k+1}^* \\ 0 \end{bmatrix}$$

and

$$x_* = V_t y_* = [V_k, v_{k+1}, V^\perp] \begin{bmatrix} R_k & \theta_{k+1} e_k & 0 \\ 0 & \rho_{k+1}^* & 0 \\ 0 & \dot{\theta}_{k+2} e_1 & \dot{L}_t \end{bmatrix}^{-1} \begin{bmatrix} f_k \\ \phi_{k+1}^* \\ 0 \end{bmatrix}. \tag{3.12}$$

By performing the QL factorization $\dot{L}_t^T = \ddot{Q}_t \ddot{L}_t$ and examining the top $(k+2)\times(k+2)$ subblock of the system in (3.12), we conclude that

$$x_* = [V_k, v_{k+1}, \ddot{v}] \begin{bmatrix} R_k & \theta_{k+1} e_k & 0 \\ 0 & \rho_{k+1}^* & 0 \\ 0 & \dot{\theta}_{k+2} & \ddot{\rho}_{k+2} \end{bmatrix}^{-1} \begin{bmatrix} f_k \\ \phi_{k+1}^* \\ 0 \end{bmatrix} \tag{3.13}$$

for some $\ddot{\rho}_{k+2}$ and unit vector $\ddot{v}$. The claim in (3.6) follows, and from (3.7) it can be seen that $\ddot{v} = -v^\perp$ and $\dot{\theta}_{k+2} = \xi\ddot{\rho}_{k+2}$. We note that without loss of generality the matrices $\dot{Q}_t$ and $\ddot{Q}_t$ may be constructed so that $\dot{\theta}_{k+2}$ and $\ddot{\rho}_{k+2}$ are nonnegative.

As for the bound in (3.9), we observe that

$$\begin{bmatrix} \ddot{R}_{k+2} \\ 0 \end{bmatrix} = \dot{Q}_t R_t \begin{bmatrix} I_{k+1} & 0 \\ 0 & \ddot{Q}_t \end{bmatrix} \begin{bmatrix} I_{k+2} \\ 0 \end{bmatrix}$$

and therefore that $\sigma_{\min}(\ddot{R}_{k+2}) \geq \sigma_{\min}(R_t) \geq \sigma_{\min}(A)$ by the Cauchy interlacing theorem (similarly, we find that $\sigma_{\min}(R_{k+1}^*) \geq \sigma_{\min}(\ddot{R}_{k+2})$).

Finally, by comparing (3.5) with (2.5) and (2.6), it can be seen that $\rho_{k+1}^*\phi_{k+1}^* = \rho'_{k+1}\phi'_{k+1}$. Since

$$|\phi_{k+1}^*| = \|\Pi_A r_k^Q\|_2 \leq \|r_k^Q\|_2 = |\phi'_{k+1}|,$$

it follows that $\rho_{k+1}^* \geq \rho'_{k+1}$. ∎

Since $\sigma_{\min}(R_{k+1}^*) \geq \sigma_{\min}(A) \geq \sigma_{\min}(\widetilde{R}_{k+1})$, it follows that $\rho_{k+1}^* \geq \tilde{\rho}_{k+1}$. This implies in turn that $|\phi_{k+1}^*| \leq |\tilde{\phi}_{k+1}|$, and so by comparing the forms for x_{k+1}^* and $\tilde{x}_{k+1}$ in (3.6) and (3.3), we get the following corollary.

COROLLARY 3.3. *x_{k+1}^* is a convex combination of x_k^Q and $\tilde{x}_{k+1}$.*

3.1.1. LDL^T factorizations. Given a nonzero lower bound $\tilde{\sigma} \leq \sigma_{\min}(A)$, we can obtain practical bounds on the unknown quantities ρ_{k+1}^*, ξ, and $\ddot{\rho}_{k+2}$. For convenience, we will work with the quantity $\dot{\theta}_{k+2} = \xi\ddot{\rho}_{k+2}$ rather than ξ for the remainder of the paper.

The inequality in (3.9) implies that $\sigma_{\min}(\ddot{R}_{k+2}) \geq \tilde{\sigma}$, so we may obtain bounds on our unknown quantities by considering the LDL^T factorization of the shifted tridiagonal matrix $\ddot{R}_{k+2}^T \ddot{R}_{k+2} - \tilde{\sigma}^2 I$ and using the fact that the elements of the diagonal matrix D must be nonnegative.

The resulting factorization will be nearly identical to the LDL^T factorization of $\widetilde{R}_{k+1}^T \widetilde{R}_{k+1} - \tilde{\sigma}^2 I$, from which the recurrence (3.4) for $\tilde{\rho}_{k+1}$ may be derived (see [16, sect. 3] for details). From the close relation between these two factorizations, we derive the constraints[6]

$$\rho_{k+1}^{*2} + \dot{\theta}_{k+2}^2 - \tilde{\rho}_{k+1}^2 \geq 0, \tag{3.14a}$$

$$\ddot{\rho}_{k+2}^2 - \tilde{\sigma}^2 - \frac{\dot{\theta}_{k+2}^2 \ddot{\rho}_{k+2}^2}{\rho_{k+1}^{*2} + \dot{\theta}_{k+2}^2 - \tilde{\rho}_{k+1}^2} \geq 0. \tag{3.14b}$$

Equivalently,

$$\begin{bmatrix} \rho_{k+1}^* & 0 \\ \dot{\theta}_{k+2} & \ddot{\rho}_{k+2} \end{bmatrix}^T \begin{bmatrix} \rho_{k+1}^* & 0 \\ \dot{\theta}_{k+2} & \ddot{\rho}_{k+2} \end{bmatrix} - \begin{bmatrix} \tilde{\rho}_{k+1}^2 & 0 \\ 0 & \tilde{\sigma}^2 \end{bmatrix} \succeq 0. \tag{3.15}$$

This last inequality is the key to proving that x_* falls within a particular ellipsoid, which we do in the following section.

3.2. An ellipsoidal bound. Recall that the set

$$\left\{(x, y) \ : \ x^2/\omega_1^2 + y^2/\omega_2^2 \leq 1\right\}$$

describes a two-dimensional ellipse with semiaxes of length ω_1 and ω_2. With this example in mind, we define the point

$$\tilde{x}_{k+1}^{\mathcal{E}} := \frac{1}{2}\left(x_k^Q + \tilde{x}_{k+1}\right) = x_k^Q + \frac{\tilde{\phi}_{k+1}}{2\tilde{\rho}_{k+1}} h_{k+1} \tag{3.16}$$

and the set

(3.17)

$$\mathcal{E}_{k+1} := \left\{ x_k^Q + \zeta_1 h_{k+1} + \zeta_2 v^\perp \ : \ \left(\frac{2\tilde{\rho}_{k+1}}{\tilde{\phi}_{k+1}}\right)^2 \left(\zeta_1 - \frac{\tilde{\phi}_{k+1}}{2\tilde{\rho}_{k+1}}\right)^2 + \left(\frac{2\tilde{\sigma}}{\tilde{\phi}_{k+1}}\right)^2 \zeta_2^2 \leq 1, \ \|v^\perp\|_2 = 1, \ V_{k+1}^T v^\perp = 0 \right\}.$$

Then $\mathcal{E}_{k+1}$ (Figure 3.1) describes an $(n-k)$-dimensional ellipsoid in $\mathbb{R}^n$ centered at $\tilde{x}_{k+1}^{\mathcal{E}}$. One axis is the line segment from x_k^Q to $\tilde{x}_{k+1}$ and has length $\frac{|\tilde{\phi}_{k+1}|}{\tilde{\rho}_{k+1}}\|h_{k+1}\|_2$, and the remainining axes are all orthogonal to V_{k+1} and have length $|\tilde{\phi}_{k+1}|/\tilde{\sigma}$. The first axis is necessarily shorter, since

$$\frac{|\tilde{\phi}_{k+1}|}{\tilde{\rho}_{k+1}}\|h_{k+1}\|_2 = |\tilde{\phi}_{k+1}|\|V_{k+1}\widetilde{R}_{k+1}^{-1} e_{k+1}\|_2 \leq \frac{|\tilde{\phi}_{k+1}|}{\tilde{\sigma}}.$$

We claim that the solution x_* must fall within this ellipsoid.

[6] Defining $0/0 = 0$ in the event that $\dot{\theta}_{k+2} = 0$ and $\rho_{k+1}^* = \tilde{\rho}_{k+1}$.

THEOREM 3.4. *Let $\tilde{\sigma}$, $\tilde{\rho}_{k+1}$, $\tilde{\phi}_{k+1}$, and $\tilde{x}_{k+1}$ be defined as in Theorem* 3.1. *Then $x_* \in \mathcal{E}_{k+1}$.*

Before proving this theorem, we introduce a useful lemma.

LEMMA 3.5. *For fixed b, the set $\{x \ : \ Cx = b,\ C \succeq I\}$ is a ball with center $b/2$ and radius $\|b/2\|_2$.*

Proof. With the substitution $D = C^{-1} - I/2$, we find that

$$\begin{aligned}\{x \ : \ Cx = b,\ C \succeq I\} &= \{b/2 + Db \ : \ -I/2 \preceq D \preceq I/2\} \\ &\subseteq \{b/2 + Db \ : \ \|D\|_2 \leq 1/2\}.\end{aligned}$$

The final set is exactly the ball described. Furthermore, the inequality is actually an equality: for any point $b/2 + z$ in the ball, we may choose $D = \kappa H$, where $\kappa = \|z\|_2/\|b\|_2$ and H is a Householder reflection that takes b to $z\|b\|_2/\|z\|_2$. For the case $z = 0$, we may use $D = 0$. ∎

We are now ready to prove the main theorem.

Proof of Theorem 3.4. Let ϕ^*_{k+1} and ρ^*_{k+1} be defined as in Lemma 3.2. Lemma 3.2 implies that the solution to (LS) may be written in the form

$$x_* = x_k^Q + \zeta_1 h_{k+1} + \zeta_2 v^\perp, \tag{3.18}$$

where

$$\begin{bmatrix} \rho^*_{k+1} & 0 \\ \dot{\theta}_{k+2} & \ddot{\rho}_{k+2} \end{bmatrix} \begin{bmatrix} \zeta_1 \\ \zeta_2 \end{bmatrix} = \begin{bmatrix} \phi^*_{k+1} \\ 0 \end{bmatrix} \tag{3.19}$$

for some nonnegative scalars $\dot{\theta}_{k+2}$ and $\ddot{\rho}_{k+2}$. By defining

$$\widehat{R}_{k+2} := \begin{bmatrix} \rho^*_{k+1}/\tilde{\rho}_{k+1} & 0 \\ \dot{\theta}_{k+2}/\tilde{\rho}_{k+1} & \ddot{\rho}_{k+2}/\tilde{\sigma} \end{bmatrix} \tag{3.20}$$

and using the fact that $\rho^*_{k+1}\phi^*_{k+1} = \tilde{\rho}_{k+1}\tilde{\phi}_{k+1}$, we may rewrite (3.19) as

$$\widehat{R}^T_{k+2}\widehat{R}_{k+2} \begin{bmatrix} \tilde{\rho}_{k+1}\zeta_1 \\ \tilde{\sigma}\zeta_2 \end{bmatrix} = \begin{bmatrix} \tilde{\phi}_{k+1} \\ 0 \end{bmatrix}.$$

By working from (3.15), we can check that $\widehat{R}^T_{k+2}\widehat{R}_{k+2} \succeq I$. By applying Lemma 3.5, it follows that the vector $\left[\begin{smallmatrix} \tilde{\rho}_{k+1}\zeta_1 \\ \tilde{\sigma}\zeta_2 \end{smallmatrix}\right]$ lies in a ball with center $\left[\begin{smallmatrix} \tilde{\phi}_{k+1}/2 \\ 0 \end{smallmatrix}\right]$ and radius $|\tilde{\phi}_{k+1}|/2$. The vector $\left[\begin{smallmatrix} \zeta_1 \\ \zeta_2 \end{smallmatrix}\right]$ therefore lies in an ellipse centered at $\left[\begin{smallmatrix} \tilde{\phi}_{k+1}/(2\tilde{\rho}_{k+1}) \\ 0 \end{smallmatrix}\right]$, having semiaxes of length $|\tilde{\phi}_{k+1}|/(2\tilde{\rho}_{k+1})$ and $|\tilde{\phi}_{k+1}|/(2\tilde{\sigma})$. By applying this bound to the formula for x_* in (3.18), the theorem follows. ∎

In the least-squares case, by taking (3.19) and using the relations $\rho^*_{k+1} \geq \rho'_{k+1}$ and $\rho^*_{k+1}\phi^*_{k+1} = \rho'_{k+1}\phi'_{k+1}$, we obtain the additional constraint

$$\zeta_1 \leq \phi'_{k+1}/\rho'_{k+1}.$$

Equivalently (compare (3.6) and (2.9)), x^*_{k+1} is a convex combination of x_k^Q and x_{k+1}^G. In this situation, x_* must lie in the intersection of an ellipsoid and a half-space, as illustrated later in Figure 3.2. More formally, we define the domain

$$\mathcal{E}^{(G)}_{k+1} := \mathcal{E}_{k+1} \cap \left\{ x_k^Q + \zeta_1 h_{k+1} + x^\perp \ : \ \zeta_1 \leq \frac{\phi'_{k+1}}{\rho'_{k+1}},\ V^T_{k+1}x^\perp = 0 \right\} \tag{3.21}$$

and get the following extension of Theorem 3.4.

THEOREM 3.6. *With all of the definitions used in Theorem* 3.4, *$x^* \in \mathcal{E}^{(G)}_{k+1}$.*

3.3. Tightness of bounds. If the only pieces of information we use to derive our error bounds are

- the information obtained by running k steps of Algorithm 2.1, and
- a lower bound $\tilde{\sigma} \leq \sigma_{\min}(A)$,

then the bound on x_* established in Theorem 3.6 is sharp. We prove this assertion in this section.

DEFINITION 3.7. *Given A, b, $\tilde{\sigma} \leq \sigma_{\min}(A)$, and a nonnegative integer k, we say that a matrix A' is* indistinguishable *from A (with respect to $(b, k, \tilde{\sigma})$) if*

- *Algorithm* 2.1 *behaves identically on the inputs (A, b) and (A', b) for the first k iterations, and*
- $\sigma_{\min}(A') \geq \tilde{\sigma}$.

THEOREM 3.8. *Fix $\tilde{\sigma} \leq \sigma_{\min}(A)$, and say that we have run k steps of Algorithm* 2.1. *For any z in the interior of $\mathcal{E}_{k+1}^{(G)}$, there exists a matrix $A^{(z)}$, indistinguishable from A, such that z is the minimum-norm solution to $\min_x \|A^{(z)}x - b\|_2$.*

Proof. We start by essentially reversing the process used to prove Theorem 3.4. Since $z \in \mathcal{E}_{k+1}^{(G)}$, the vector z may be written in the form

$$z = x_k^Q + \zeta_1 h_{k+1} + \zeta_2 v^\perp \tag{3.22}$$

for some unit vector $v^\perp$ such that $V_{k+1}^T v^\perp = 0$. Without loss of generality, we may set ζ_2 and $v^\perp$ so that ζ_1 and ζ_2 have opposite signs. Furthermore, ζ_1 and ζ_2 satisfy

$$\begin{bmatrix} \tilde{\rho}_{k+1}\zeta_1 \\ \tilde{\sigma}\zeta_2 \end{bmatrix} = \begin{bmatrix} \tilde{\phi}_{k+1}/2 \\ 0 \end{bmatrix} + D^{(z)} \begin{bmatrix} \tilde{\phi}_{k+1} \\ 0 \end{bmatrix}$$

for some symmetric matrix $D^{(z)}$ (not necessarily unique) such that $\|D^{(z)}\|_2 < 1/2$. This inequality is strict because of the assumption that z is in the interior of $\mathcal{E}_{k+1}^{(G)}$, and guarantees that $C^{(z)} := (D^{(z)} + I/2)^{-1}$ is well-defined. It follows that

$$C^{(z)} \begin{bmatrix} \tilde{\rho}_{k+1}\zeta_1 \\ \tilde{\sigma}\zeta_2 \end{bmatrix} = \begin{bmatrix} \tilde{\phi}_{k+1} \\ 0 \end{bmatrix}$$

for some matrix $C^{(z)} \succeq I$.

Since $C^{(z)}$ is positive definite, there exists a lower triangular matrix $\widehat{R}^{(z)}$ such that $C^{(z)} = \widehat{R}^{(z)T}\widehat{R}^{(z)}$. From there, there exist quantities $\rho_{k+1}^{*(z)}$, $\dot{\theta}_{k+2}^{(z)}$, and $\ddot{\rho}_{k+2}^{(z)}$ so that $\widehat{R}^{(z)}$ has the form specified in (3.20). Without loss of generality, these quantities are nonnegative. In particular, if $\ddot{\rho}_{k+2}$ is nonnegative and ζ_1 and ζ_2 are chosen to have opposite signs, then (3.19) implies that $\ddot{\theta}_{k+2}^{(z)}$ is nonnegative as well. We use these quantities to define a matrix $\ddot{R}_{k+2}^{(z)}$ having the form specified by (3.8). The vector z then satisfies the equation

$$z = [V_{k+1}, -v^\perp]\ddot{R}_{k+2}^{(z)-1} \begin{bmatrix} f_k \\ \phi_{k+1}^* \\ 0 \end{bmatrix}, \tag{3.23}$$

paralleling (3.13).

Since $z \in \mathcal{E}_{k+1}^{(G)}$, it follows that $\zeta_1 \leq \phi'_{k+1}/\rho'_{k+1}$ and thus by (3.19) that $\rho_{k+1}^{*(z)} \geq \rho'_{k+1}$. This inequality in turn implies that there exists a lower bidiagonal matrix $B_{k+2}^{(z)}$

whose leading entries are given by L_{k+1} such that the QR factorizations of $B_{k+2}^{(z)}$ and $\ddot{R}_{k+2}^{(z)}$ yield the same R factor. We omit the details, but this assertion may be verified by using a few plane rotations.

Finally, we construct $A^{(z)}$ to have the form

$$A^{(z)} = U_{k+3} B_{k+2}^{(z)} V_{k+2}^{(z)T},$$

where $V_{k+2}^{(z)} = [V_{k+1}, -v^{\perp}]$, and U_{k+3} is any orthogonal matrix extending U_{k+1}. The matrix $A^{(z)}$ has the following properties:

- Algorithm 2.1 behaves identically on the inputs (A, b) and $(A^{(z)}, b)$ for the first k iterations, returning $(U_{k+1}, V_{k+1}, L_{k+1})$.
- $\sigma_{\min}(A^{(z)}) = \sigma_{\min}(B_{k+2}^{(z)}) = \sigma_{\min}(\ddot{R}_{k+2}^{(z)})$. Since $C^{(z)} \succeq 1$ it follows from section 3.1.1 that $\sigma_{\min}(\ddot{R}_{k+2}^{(z)}) \geq \tilde{\sigma}$ and therefore that $A^{(z)}$ and A are indistinguishable.
- From (3.23) it follows that if we run LSQR on $(A^{(z)}, b)$, the algorithm will terminate in at most $k+2$ iterations and return z. Thus z is the minimum-norm solution to $\min_x \|A^{(z)}x - b\|_2$. ∎

The proof that $x_* \in \mathcal{E}_{k+1}$ for (SPD) is essentially the same, except that we construct the matrix

$$\bar{A}^{(z)} = V_{k+2}^{(z)} T_{k+2}^{(z)} V_{k+2}^{(z)T},$$

where $T_{k+2}^{(z)} := \ddot{R}_{k+2}^{(z)T} \ddot{R}_{k+2}^{(z)}$, and the constraint $\rho_{k+1}^{*} \geq \rho_{k+1}'$ no longer applies.

3.4. Minimizing the error bound. By Theorem 3.4, x_* falls somewhere in the ellipsoid $\mathcal{E}_{k+1}$. In the least-squares case, Theorem 3.6 tells us that x_{k+1}^{*} also lies between x_k^Q and x_{k+1}^G. If $\|x_{k+1}^G\|_2 \leq \|\tilde{x}_{k+1}\|_2$, then we can tighten the bound on x_*. Figures 3.1 and 3.2 illustrate these bounds.

By Theorem 3.8, unless we acquire additional information about our problem the solution x_* could lie anywhere in the interior of $\mathcal{E}_{k+1}^{(G)}$. For a generic point x_{k+1}, we therefore propose the error bound

$$\|x_{k+1} - x_*\|_2 \leq \max_{z \in \mathcal{E}_{k+1}^{(G)}} \|x_{k+1} - z\|_2. \tag{3.24}$$

We can cheaply solve this maximization problem for LSQR (or for CG, using $\mathcal{E}_{k+1}$ instead of $\mathcal{E}_{k+1}^{(G)}$), but the resulting bound is not particularly elegant. It is more natural to find the point for which the right-hand side of (3.24) is minimized, and for (SPD)

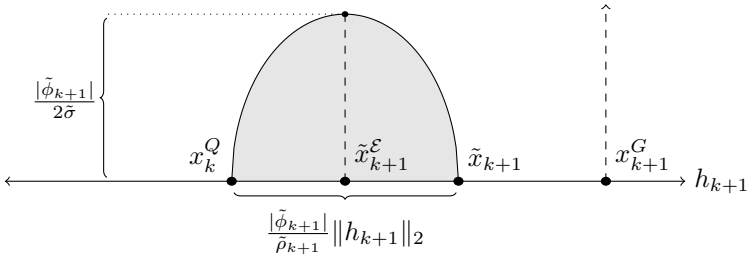

FIG. 3.1. *The center of the ellipsoid $\tilde{x}_{k+1}^{\mathcal{E}}$ often minimizes the error bound.*

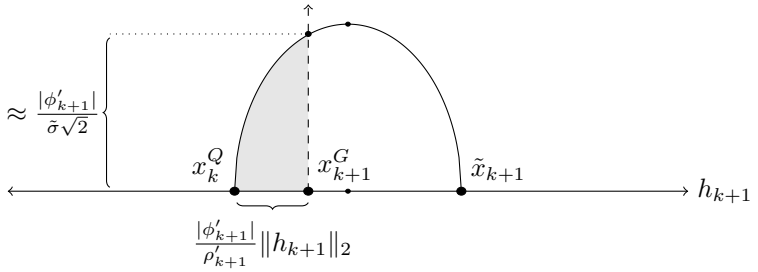

FIG. 3.2. *The Craig iterate x_{k+1}^G minimizes the error bound whenever $\frac{|\phi'_{k+1}|}{\rho'_{k+1}} \leq \frac{|\tilde{\phi}_{k+1}|}{2\tilde{\rho}_{k+1}}$.*

this point is always $\tilde{x}_{k+1}^{\mathcal{E}}$. For (LS) it is either $\tilde{x}_{k+1}^{\mathcal{E}}$ or x_{k+1}^G, whichever has the smaller norm (we denote this point $\tilde{x}_{k+1}^{\mathcal{E}(G)}$).

The point $\tilde{x}_{k+1}^{\mathcal{E}}$ always satisfies the bound

$$\left\|\tilde{x}_{k+1}^{\mathcal{E}} - x_*\right\|_2 \leq \frac{|\tilde{\phi}_{k+1}|}{2\tilde{\sigma}}, \tag{3.25}$$

which is precisely a factor of 2 better than the earlier bound (1.1) for $\|x_k^Q - x_*\|_2$.

If $\|x_{k+1}^G\|_2 \leq \|\tilde{x}_{k+1}^{\mathcal{E}}\|_2$, then the bound

$$\left\|x_{k+1}^G - x_*\right\| \leq \max_{z \in \mathcal{E}_{k+1}^{(G)}} \left\|x_{k+1}^G - z\right\|_2$$

is maximized when the projection of z onto $\mathrm{Span}(V_{k+1})$ is x_{k+1}^G (see Figure 3.2). By defining

$$\tilde{c}_{k+1} := \frac{\rho'_{k+1}}{\tilde{\rho}_{k+1}} = \frac{\tilde{\phi}_{k+1}}{\phi'_{k+1}} \tag{3.26}$$

and working from (3.22) and the definition of $\mathcal{E}_{k+1}$ in (3.17), we obtain after some algebra the bound

$$\left\|x_{k+1}^G - x_*\right\| \leq \frac{|\phi'_{k+1}|}{\tilde{\sigma}}\sqrt{1 - \tilde{c}_{k+1}^{-2}}. \tag{3.27}$$

But if $\|x_{k+1}^G\|_2 \leq \|\tilde{x}_{k+1}^{\mathcal{E}}\|_2$, then $\tilde{c}_{k+1}^{-2} \leq 1/2$, and the bound in (3.27) is at most a factor of $\sqrt{2}$ better than the trivial bound

$$\left\|x_k^Q - x_*\right\|_2 \leq \frac{\|r_k^Q\|_2}{\tilde{\sigma}} = \frac{|\phi'_{k+1}|}{\tilde{\sigma}}.$$

The bound (3.27) also applies if we happen to know that the system $Ax = b$ is consistent, in which case Craig's method is optimal and $x_{k+1}^* = x_{k+1}^G$. Such a situation might arise if A has more columns than rows and we wish to find the minimum-norm x such that $Ax = b$. For more information on this topic, see [3].

3.5. New error bounds for LSQR and CG. Here we derive the error bounds for LSQR and CG implied by (3.24), which are to the best of our knowledge novel.

We define the quantities

$$\omega_1 := \frac{|\tilde{\phi}_{k+1}|}{2\tilde{\rho}_{k+1}}\|h_{k+1}\|_2 \quad \text{and} \quad \omega_2 := \frac{|\tilde{\phi}_{k+1}|}{2\tilde{\sigma}},$$

which are the lengths of the semiaxes of $\mathcal{E}_{k+1}$. Slightly modifying the parameterization used for $\mathcal{E}_{k+1}$ in (3.17), we write the solution in the form

$$x_* = x_k^Q + \zeta_1 \frac{h_{k+1}}{\|h_{k+1}\|_2} + \zeta_2 v^\perp$$

for some unit vector $v^\perp$ orthogonal to V_{k+1}, where ζ_1 and ζ_2 satisfy the bound

$$\frac{1}{\omega_1^2}(\zeta_1 - \omega_1)^2 + \frac{1}{\omega_2^2}\zeta_2^2 \le 1.$$

Therefore,

$$\left\|x_k^Q - x_*\right\|_2 = (\zeta_1^2 + \zeta_2^2)^{1/2} \le \left(\zeta_1^2 + \omega_2^2 - \frac{\omega_2^2}{\omega_1^2}(\zeta_1 - \omega_1)^2\right)^{1/2} =: f_k(\zeta_1).$$

Since $\zeta_1 \in [0, \min\{1, \tilde{c}_{k+1}^{-1}\}]$ for (LS) and $\zeta_1 \in [0,1]$ for (SPD), we may write our new error bound for LSQR as

$$\|x_k^Q - x_*\|_2 \le \max_{\zeta \in [0, \min\{1, \tilde{c}_{k+1}^{-1}\}]} f_k(\zeta), \tag{3.28}$$

and similarly for CG, but with $\zeta \in [0,1]$.

The maximization problems for LSQR and CG have the respective solutions

$$\tilde{\zeta}_k := \min\left\{1, \tilde{c}_{k+1}^{-2}, \frac{\omega_2^2}{\omega_2^2 - \omega_1^2}\right\} \quad \text{and} \quad \tilde{\zeta}_k := \min\left\{1, \frac{\omega_2^2}{\omega_2^2 - \omega_1^2}\right\}.$$

The resulting LSQR bound will satisfy the inequalities

$$\min\left\{\frac{|\tilde{\phi}_{k+1}|}{2\tilde{\sigma}}, \frac{|\phi'_{k+1}|}{\tilde{\sigma}\sqrt{2}}\right\} \le f_k(\tilde{\zeta}_k) \le \min\left\{\frac{|\tilde{\phi}_{k+1}|}{\tilde{\sigma}}, \frac{|\phi'_{k+1}|}{\tilde{\sigma}}\right\}, \tag{3.29}$$

and the CG bound will satisfy the inequalities

$$\frac{|\tilde{\phi}_{k+1}|}{2\tilde{\sigma}} \le f_k(\tilde{\zeta}_k) \le \frac{|\tilde{\phi}_{k+1}|}{\tilde{\sigma}}. \tag{3.30}$$

Since our error bound (3.25) for $\tilde{x}_{k+1}^{\mathcal{E}}$ is equal to the left-hand side of (3.30) and our error bound for $\tilde{x}_{k+1}^{\mathcal{E}(G)}$ (i.e., the minimum of (3.25) and (3.27)) is at least as great as the left-hand side of (3.29), we conclude that our optimal bounds from the previous section outperform our bounds for CG and LSQR by at most a factor of 2.

3.6. Monotonicity results. The tightness result of Theorem 3.8 shows that $\mathcal{E}_{k+1}^{(G)}$ is the smallest region that provably contains x_* given $\tilde{\sigma}$ and the information available to us after k iterations of Algorithm 2.1. Since we get more information with each new iteration, the sets $\mathcal{E}_{k+1}^{(G)}$ must shrink accordingly.

COROLLARY 3.9 (corollary to Theorem 3.8). *For all $k \ge 1$, $\mathcal{E}_{k+1}^{(G)} \subseteq \mathcal{E}_k^{(G)}$. Similarly, $\mathcal{E}_{k+1} \subseteq \mathcal{E}_k$.*

This corollary leads immediately to several nice (though not necessarily novel) monotonicity results.

THEOREM 3.10. *For fixed $\tilde{\sigma} \le \sigma_{\min}(A)$, the following hold in exact arithmetic:*

(a) $\|\tilde{x}_{k+1}\|_2$ *decreases monotonically.*
(b) *The bound* $\|x_k^Q - x_*\|_2 \le (\|\tilde{x}_{k+1}\|_2^2 - \|x_k^Q\|_2^2)^{1/2}$ *decreases monotonically.*
(c) $|\tilde{\phi}_{k+1}|$ *decreases monotonically.*
(d) *The bound* $\|x_k^Q - x_*\|_2 \le \min\{\frac{|\tilde{\phi}_{k+1}|}{\tilde{\sigma}}, \frac{|\phi'_{k+1}|}{\tilde{\sigma}}\}$ *decreases monotonically.*
(e) *The bound* $\|x_k^Q - x_*\|_2 \le f_k(\tilde{\zeta}_k)$ *decreases monotonically.*
(f) *The bound*

$$\|\tilde{x}_{k+1}^{\mathcal{E}(G)} - x_*\|_2 \le \begin{cases} \frac{|\tilde{\phi}_{k+1}|}{2\tilde{\sigma}}, & \tilde{c}_{k+1}^{-2} > 1/2, \\ \frac{|\phi'_{k+1}|}{\tilde{\sigma}}\sqrt{1-\tilde{c}_{k+1}^{-2}}, & \tilde{c}_{k+1}^{-2} \le 1/2, \end{cases}$$

decreases monotonically.
(g) $\min\{\tilde{c}_{k+1}, 1\}$ *decreases monotonically.*

Proof.
(a) By Theorem 3.1, $\|x_*\|_2 \le \|\tilde{x}_{k+1}\|_2$. Thus $\|\tilde{x}_{k+1}\|_2$ is the point in $\mathcal{E}_{k+1}$ with the largest norm. Since $\mathcal{E}_{k+1} \subseteq \mathcal{E}_k$, $\|\tilde{x}_{k+1}\|_2 \le \|\tilde{x}_k\|_2$.
(b) $\|\tilde{x}_{k+1}\|_2$ decreases monotonically and $\|x_k^Q\|_2$ increases monotonically.
(c) $\mathcal{E}_{k+1}$ has diameter $|\tilde{\phi}_{k+1}|/\tilde{\sigma}$.
(d) $|\tilde{\phi}_{k+1}|$ decreases monotonically, and so does $|\phi'_{k+1}| = \|r_k^Q\|_2$.
(e) x_k^Q is the point in $\mathcal{E}_{k+1}^{(G)} \cap \mathrm{Span}(V_{k+1})$ that maximizes the bound (3.24), but x_{k+1}^Q is also in $\mathcal{E}_{k+1}^{(G)} \cap \mathrm{Span}(V_{k+1})$. Thus

$$\max_{z \in \mathcal{E}_{k+1}^{(G)}} \left\| x_k^Q - z \right\|_2 \ge \max_{z \in \mathcal{E}_{k+1}^{(G)}} \left\| x_{k+1}^Q - z \right\|_2 \ge \max_{z \in \mathcal{E}_{k+2}^{(G)}} \left\| x_{k+1}^Q - z \right\|_2.$$

(f) Since $\mathrm{Span}(V_{k+1}) \subseteq \mathrm{Span}(V_{k+2})$ and $\mathcal{E}_{k+2}^{(G)} \subseteq \mathcal{E}_{k+1}^{(G)}$, it follows that

$$\min_{x_{k+1} \in \mathrm{Span}(V_{k+1})} \max_{z \in \mathcal{E}_{k+1}^{(G)}} \|x_{k+1} - z\|_2 \ge \min_{x_{k+2} \in \mathrm{Span}(V_{k+2})} \max_{z \in \mathcal{E}_{k+2}^{(G)}} \|x_{k+2} - z\|_2.$$

(g) Since the residuals in LSQR are updated along orthogonal directions, it can be seen that

$$\left\| \Pi_A r_k^Q \right\|_2^2 = \left\| \Pi_A r_{k-1}^Q \right\|_2^2 - \left\| r_k^Q - r_{k-1}^Q \right\|_2^2 \le \tilde{\phi}_k^2 - \phi_k^2.$$

But $|\tilde{\phi}_{k+1}|$ is the tightest available bound on $\|\Pi_A r_k^Q\|_2$ after k iterations, so $\tilde{\phi}_{k+1}^2 \le \tilde{\phi}_k^2 - \phi_k^2$. Since $\phi_{k+1}'^2 = \phi_k'^2 - \phi_k^2$, it follows that if $\tilde{c}_k \le 1$, then

$$\tilde{c}_{k+1}^2 = \tilde{\phi}_{k+1}^2 / \phi_{k+1}'^2 < \tilde{\phi}_k^2 / \phi_k'^2 = \tilde{c}_k^2. \qquad \square$$

4. Practical considerations. As with earlier error bounds, our bounds depend on finding the quantity $\tilde{\rho}_{k+1}$ such that $\sigma_{\min}(\widetilde{R}_{k+1}) = \tilde{\sigma}$, where $\widetilde{R}_{k+1}$ was defined in (3.1). As was observed in [4] and [23], the quality of the estimate $\tilde{\rho}_{k+1}$ is sensitive to the choice of $\tilde{\sigma}$. Too small a value of $\tilde{\sigma}$ can lead to weak bounds, but due to the cancellation involved in (3.4), if $\tilde{\sigma}$ is too large, then the procedure for computing $\tilde{\rho}_{k+1}$ may fail entirely, *sometimes even if* $\tilde{\sigma}$ *is smaller than* $\sigma_{\min}(A)$. Thus even if we are fortunate enough to know $\sigma_{\min}(A)$ precisely, it is prudent to choose $\tilde{\sigma}$ to be somewhat smaller. The authors of [4] suggest using an estimate such as $\tilde{\sigma} = \sigma_{\min}(A)(1 - 10^{-10})$.

For many practical problems, we will not have access to a lower bound $\tilde{\sigma}$. We could therefore consider estimating $\tilde{\sigma}$ at each iteration by $\sigma_{\min}(R_k)$ or an approximation thereof (several methods for doing so are discussed in [17]) and using it with an error estimate that is less sensitive to the choice of $\tilde{\sigma}$. Instead of choosing $\tilde{\rho}_{k+1}$ so that $\sigma_{\min}(\widetilde{R}_{k+1}) = \tilde{\sigma}$, for example, we could choose $\tilde{\rho}_{k+1}$ so when we consider the QR factorization of $\widetilde{R}_{k+1}^T$, the final entry in the R factor is equal to $\tilde{\sigma}$. With this strategy, the error bound in (1.1) simplifies to

$$\left\|x_k^Q - x_*\right\|_2 < \frac{\|A^T r_k^M\|_2}{\tilde{\sigma}^2} = \frac{\|\bar{r}_k^M\|_2}{\tilde{\sigma}^2}, \tag{4.1}$$

where r_k^M is the residual from LSMR [6] (equivalent to MINRES [19] on the normal equations), which chooses $x_k^M \in \text{Span}(V_k)$ to minimize $\|A^T r_k^M\|_2$, and $\bar{r}_k^M$ is the residual from running MINRES on (SPD). Meurant and Tichý [17, Thm. 1] have proposed an estimate that is equivalent to this one in exact arithmetic but which also holds (up to minor inaccuracy) in finite precision, and which can be computed at $\mathcal{O}(1)$ cost per iteration. Although it is a loose bound, it is both monotonically decreasing and relatively insensitive to the choice of $\tilde{\sigma}$. We note that if we estimate $\tilde{\sigma}$ by $\sigma_{\min}(R_k)$, then our error estimates are no longer guaranteed to be upper bounds, but given enough time for $\sigma_{\min}(R_k)$ to converge to $\sigma_{\min}(A)$ they may be close enough for practical applications.

5. Implications for the conjugate gradient method. Our results are valid when running the CG method on any system $\bar{A}x = \bar{b}$ where $\bar{A}$ is an SPD matrix. Although CG can be derived from the Lanczos process as in section 2.6, it can also be run without forming the tridiagonal matrix T_k explicitly, as shown in Algorithm 5.1. As noted in [17, sect. 2], the quantities γ_k and δ_{k+1} relate to the entries of R_k via the equalities $\rho_k = 1/\sqrt{\gamma_k}$ and $\theta_{k+1} = \sqrt{\delta_{k+1}}/\sqrt{\gamma_k}$.

Given $\tilde{\sigma} \le \sqrt{\lambda_{\min}(\bar{A})}$, the bound from (1.1) may be computed as

$$\left\|x_k^{CG} - x_*\right\|_2 \le \frac{\tilde{\gamma}_{k+1}^{1/2}}{\tilde{\sigma}} \left\|\bar{r}_k^{CG}\right\|_2,$$

where

$$\tilde{\gamma}_1 = \frac{1}{\tilde{\sigma}^2} \quad \text{and} \quad \tilde{\gamma}_{k+1} = \frac{\tilde{\gamma}_k - \gamma_k}{\tilde{\sigma}^2(\tilde{\gamma}_k - \gamma_k) + \delta_{k+1}}.$$

Several of our main results may be summarized in the form of the following theorem.

Algorithm 5.1 Conjugate gradient method

Require: $\bar{A}, \bar{b}$
1: $x_0^{CG} = 0$
2: $\bar{r}_0^{CG} = \bar{b}$
3: $\bar{p}_1 = \bar{r}_0^{CG}$
4: **for** $k = 1, 2, \ldots$ **do**
5: $\quad \gamma_k = \frac{(\bar{r}_{k-1}^{CG})^T \bar{r}_{k-1}^{CG}}{\bar{p}_k^T \bar{A} \bar{p}_k}$
6: $\quad x_k^{CG} = x_{k-1}^{CG} + \gamma_k \bar{p}_k$
7: $\quad \bar{r}_k^{CG} = \bar{r}_{k-1}^{CG} - \gamma_k A \bar{p}_k$
8: $\quad \delta_{k+1} = \frac{(\bar{r}_k^{CG})^T \bar{r}_k^{CG}}{(\bar{r}_{k-1}^{CG})^T \bar{r}_{k-1}^{CG}}$
9: $\quad \bar{p}_{k+1} = \bar{r}_k^{CG} + \delta_{k+1} \bar{p}_k$
10: **end for**

THEOREM 5.1. *Define*

$$\tilde{x}_k^{\mathcal{E}} := x_k^{CG} + \frac{\tilde{\gamma}_{k+1}}{2}\bar{p}_{k+1}.$$

Then $x_* = \bar{A}^{-1}\bar{b}$ *lies in an ellipsoid with center* $\tilde{x}_k^{\mathcal{E}}$ *and axes of length* $\tilde{\gamma}_{k+1}\|\bar{p}_{k+1}\|_2$ *and* $\frac{\tilde{\gamma}_{k+1}^{1/2}}{\tilde{\sigma}}\|\bar{r}_k^{CG}\|_2$. *In particular,*

$$\left\|\tilde{x}_k^{\mathcal{E}} - x_*\right\|_2 \leq \frac{\tilde{\gamma}_{k+1}^{1/2}}{2\tilde{\sigma}}\left\|\bar{r}_k^{CG}\right\|_2.$$

These bounds are tight in the same sense as in Theorem 3.8.

6. Regularization. Here we extend our results to the regularized least-squares problem

$$\min_x \left\| \begin{bmatrix} A \\ \lambda I \end{bmatrix} x - \begin{bmatrix} b \\ 0 \end{bmatrix} \right\|_2,$$

where $\lambda \in \mathbb{R}$. This will allow us to find error bounds in cases where a lower bound $\tilde{\sigma}$ to $\sigma_{\min}(A)$ is unavailable, but we can also handle cases where $\tilde{\sigma} > 0$ and $\lambda > 0$ simultaneously. One possible method would be to define $\hat{A} = \left[\begin{smallmatrix} A \\ \lambda I \end{smallmatrix}\right]$ and $\hat{b} = \left[\begin{smallmatrix} b \\ 0 \end{smallmatrix}\right]$ and solve $\min_x \|\hat{A}x - \hat{b}\|_2$ using the same methods as before, but it is more efficient to take advantage of the simple structure of the regularization term λI.

In an extension of the relations in (2.1), the bidiagonalization process will now be characterized by the relations

$$\begin{bmatrix} A \\ \lambda I \end{bmatrix} V_k = \begin{bmatrix} U_{k+1} & 0 \\ 0 & V_k \end{bmatrix} \begin{bmatrix} B_k \\ \lambda I_k \end{bmatrix}, \quad \begin{bmatrix} A \\ \lambda I \end{bmatrix}^T \begin{bmatrix} U_{k+1} & 0 \\ 0 & V_k \end{bmatrix} = V_{k+1} \begin{bmatrix} B_k & \alpha_{k+1}e_{k+1} \\ \lambda I_k & 0 \end{bmatrix}^T.$$

We may perform a QR factorization $\hat{Q}_k \left[\begin{smallmatrix} B_k \\ \lambda I \end{smallmatrix}\right] = \left[\begin{smallmatrix} \hat{R}_k \\ 0 \end{smallmatrix}\right]$ following the procedure

$$\left[\begin{array}{cc:c} \hat{\rho}_k'' & 0 & \hat{\phi}_k'' \\ \beta_{k+1} & \alpha_{k+1} & 0 \\ \hline \lambda & 0 & 0 \\ 0 & \lambda & 0 \end{array}\right] \mapsto \left[\begin{array}{cc:c} \hat{\rho}_k' & 0 & \hat{\phi}_k' \\ \beta_{k+1} & \alpha_{k+1} & 0 \\ \hline 0 & 0 & * \\ 0 & \lambda & 0 \end{array}\right] \mapsto \left[\begin{array}{cc:c} \hat{\rho}_k & \hat{\theta}_{k+1} & \hat{\phi}_k \\ 0 & \hat{\rho}_{k+1}'' & \hat{\phi}_{k+1}'' \\ \hline 0 & 0 & * \\ 0 & \lambda & 0 \end{array}\right],$$

where $\hat{\rho}_1'' = \alpha_1$, as is done in [6], and $\hat{\phi}_{k+1}'' = \|b\|_2$. In this manner, the matrix $\hat{Q}_k$ alternates between operating on rows $(j, j+k+1)$ and $(j, j+1)$ for $1 \leq j \leq k$. We may define the LSQR and Craig iterates by

$$x_k^Q = V_k \hat{R}_k^{-1} \hat{f}_k \quad \text{and} \quad x_{k+1}^G = V_{k+1} \hat{R}_{k+1}'^{-1} \hat{f}_{k+1}', \tag{6.1}$$

where $\hat{R}_{k+1}'$ and $\hat{f}_{k+1}'$ have $\hat{\rho}_{k+1}'$ and $\hat{\phi}_{k+1}'$ as their final elements. The iterate x_k^Q is equivalent to the one produced by running LSQR on $\min_x \|\hat{A}x - \hat{b}\|_2$. The iterate x_{k+1}^G is not, but is instead equivalent to the iterate produced by the extended Craig method from section 2.5. The Craig iterates will no longer update along orthogonal directions, but they do allow us to derive tighter error bounds than we would have gotten by running Craig's method directly on $\min_x \|\hat{A}x - \hat{b}\|_2$. This does not contradict the tightness result of Theorem 3.8 because we know more about the structure of $\hat{A}$ than just the bound $\sqrt{\tilde{\sigma}^2 + \lambda^2} \leq \sigma_{\min}(\hat{A})$.

Aside from the change to Craig's method, our major theorems follow more or less as before. Given a bound $\tilde{\sigma} \le \sigma_{\min}(A)$, we are then left with the problem of finding the smallest positive $\check{\rho}_{k+1}$ such that

$$\sigma_{\min} \begin{bmatrix} \hat{R}_k & \hat{\theta}_{k+1} e_k \\ 0 & \check{\rho}_{k+1} \end{bmatrix} \ge \sqrt{\tilde{\sigma}^2 + \lambda^2}.$$

We could compute $\check{\rho}_{k+1}$ by the same recurrence as before, but there is a more stable method. Since there exists an orthogonal matrix $\check{Q}$ such that

$$\check{Q} \begin{bmatrix} R_k & \theta_{k+1} e_k \\ 0 & \tilde{\rho}_{k+1} \\ \lambda I & 0 \\ 0 & \lambda \end{bmatrix} = \begin{bmatrix} \hat{R}_k & \hat{\theta}_{k+1} e_k \\ 0 & \check{\rho}_{k+1} \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

we may compute $\tilde{\rho}_{k+1}$ according to (3.4) and use it with the QR factorization above to find $\check{\rho}_{k+1}$. As long as we keep track of both R_k and $\hat{R}_k$, we do not need to compute the plane rotations explicitly. Instead, we may use the recurrence

$$\hat{\lambda}_{k+1} = \left(\lambda^2 + \hat{\lambda}_k^2 \frac{\theta_{k+1}^2}{\hat{\rho}_k^2} \right)^{1/2},$$

where $\hat{\lambda}_1 = \lambda$, and at the final step compute

$$\check{\rho}_{k+1} = \left(\tilde{\rho}_{k+1}^2 + \hat{\lambda}_{k+1}^2 \right)^{1/2}.$$

In cases where $\tilde{\sigma} = 0$ we have $\tilde{\rho}_{k+1} = 0$ for (SPD) and $\tilde{\rho}_{k+1} = \rho'_{k+1}$ for (LS). Either way, and in contrast to cases where a nontrivial bound $\tilde{\sigma} \le \sigma_{\min}(A)$ is used, $\check{\rho}_{k+1}$ may be computed with no risk of breakdown.

7. Numerical experiments. We ran tests on matrices from the SuiteSparse Matrix Collection [1]. For overdetermined problems, we generally followed the procedure of Fong and Saunders [6]: problems were downloaded from the `LPnetlib` group in MATLAB, and a sparse matrix A and vector b were generated by the commands `A = (Problem.A)'` and `b = (Problem.aux.c)`, scaling b to have unit norm. Instead of removing the cases with $b = 0$ or $A^T b = 0$, we generated a vector b for the $m \times n$ matrix A according to the procedure

1. `x = (1:n)';`
2. `b = A*x + 1e-5*randn(m,1)*norm(x)*sqrt(m/(m-n));`

We considered cases with $\max\{m, n\} \le 6000$ and computed $\sigma_{\min}(A)$ and used $\tilde{\sigma} = (1 - \epsilon)\sigma_{\min}(A)$ for various values of ϵ (with $\epsilon = 10^{-10}$ being the default). We then computed x_* as `A\b` and ran all problems until the condition $\|A^T r_k^M\|_2 / (\tilde{\sigma}^2 \|x_k^Q\|_2) \le 10^{-10}$ was satisfied, using a loose but robust bound resulting from (4.1).

We measured the true errors for three sets of iterates:

- the LSQR iterates x_k^Q,
- whichever of $\tilde{x}_{k+1}^{\mathcal{E}}$ and x_{k+1}^G had smaller norm (denoted $\tilde{x}_{k+1}^{\mathcal{E}(G)}$), and
- whichever of $\tilde{x}_{k+1}$ and x_{k+1}^G had smaller norm (denoted $\tilde{x}_{k+1}^{(G)}$).

The iterate x_k^Q is the point in $\mathcal{E}_{k+1}^{(G)}$ with the smallest norm, $\tilde{x}_{k+1}^{\mathcal{E}(G)}$ minimizes our error bound (3.24), and $\tilde{x}_{k+1}^{(G)}$ has the largest norm of all points in $\mathcal{E}_{k+1}^{(G)} \cap \mathrm{Span}(V_{k+1})$.

For the first set of experiments, we compared the true errors with three error bounds: the LSLQ-based bound $\|x_k^Q - x_*\|_2 \le (\|\tilde{x}_{k+1}\|_2^2 - \|x_k^Q\|_2^2)^{1/2}$ from [4, eq.

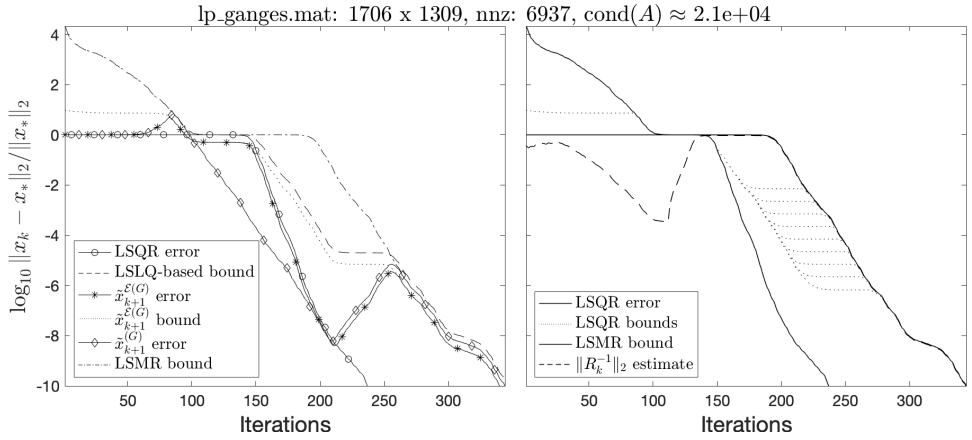

FIG. 7.1. *A relatively well-conditioned case. Left: The error increases in two distinct phases. Right: The quality of the estimate eventually degrades. Given a sufficient number of iterations, estimating* $\|R_k^{-1}\|_2$ *can potentially work well in practice.*

(36)],[7] the LSMR-based bound $\|x_k^Q - x_*\|_2 \leq \|A^T r_k^M\|_2/\tilde{\sigma}^2$ from [17], and the optimal bound on $\|\tilde{x}_{k+1}^{\mathcal{E}(G)} - x_*\|_2$ being the smaller of the bounds from (3.25) and (3.27). In general, we observed the following behavior.

1. In many cases, all three errors were nearly identical and monotonically decreasing after the first few iterations. The error bounds were all close and generally within an order of magnitude of the true errors. A second type of pattern (see Figure 7.1) was more interesting. While the LSQR error always decreased monotonically, the errors for the other two points would increase in two different phases. The first phase would come just before $|\tilde{\phi}_{k+1}|$ dropped below $|\phi'_{k+1}|$, while the system was still plausibly consistent. In this phase, the iterates produced by Craig's method are beginning to diverge significantly but we are not yet able to determine that they are doing so.
   In the second phase, the error starts to increase right when our error bounds plateau. Here, the quality of our estimates degrade to the point where our new error bounds are barely tighter than the LSMR-based bound (4.1). We do not have an intuitive explanation for why the errors should therefore *increase* rather than merely plateau, but we will note a potentially related fact: although $\|x_k^Q\|_2$ increases and $\|\tilde{x}_{k+1}\|_2$ decreases monotonically, $\|x_k^Q - \tilde{x}_{k+1}\|_2$ (i.e., the length of the minor axis of $\mathcal{E}_{k+1}$) does not necessarily decrease monotonically.
2. Our error *bound* for $\tilde{x}_{k+1}^{\mathcal{E}(G)}$ was monotonically decreasing and always smaller than the LSLQ-based bound for x_k^Q, though never by much. Which method had the smallest true error could vary over time (see Figure 7.2), but at convergence it was generally LSQR.

For the second set of experiments, we computed the LSQR error bound (3.28) for estimates $\tilde{\sigma} = (1 - 10^{-m})\sigma_{\min}(A)$ with $4 \leq m \leq 12$. We compared these bounds to the estimate $\|A^T r_k^M\|_2/\sigma_{\min}^2(R_k)$,[8] where we used an approximation to $\sigma_{\min}(R_k)$

[7] Given $\tilde{\rho}_{k+1}$, this expression can be evaluated cheaply and without cancellation—see [4, eq. (19)] and [10, sect. 3.1].

[8] Here we used a cheap estimate of $\|A^T r_k^M\|$ instead of computing $A^T r_k^M$ explicitly.

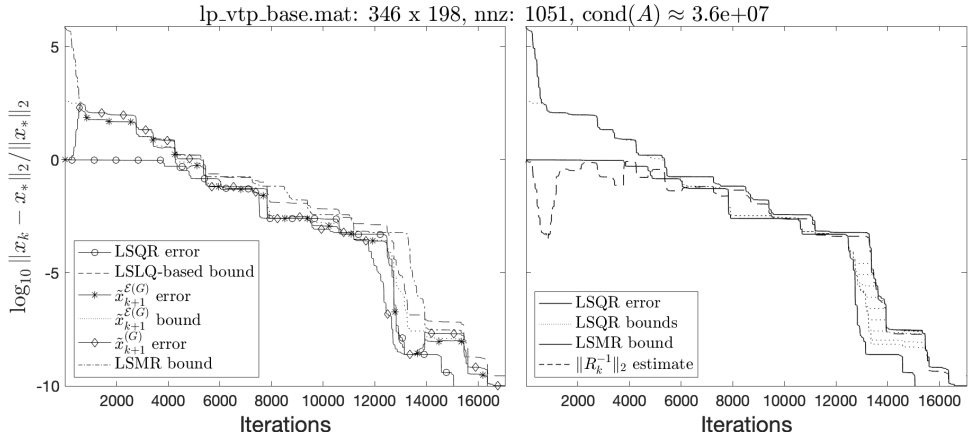

FIG. 7.2. *An ill-conditioned case. Left: We do not know which method has the smallest error at any point in time, but at convergence it was generally LSQR. Right: Our estimates still work as upper bounds despite the large number of iterations.*

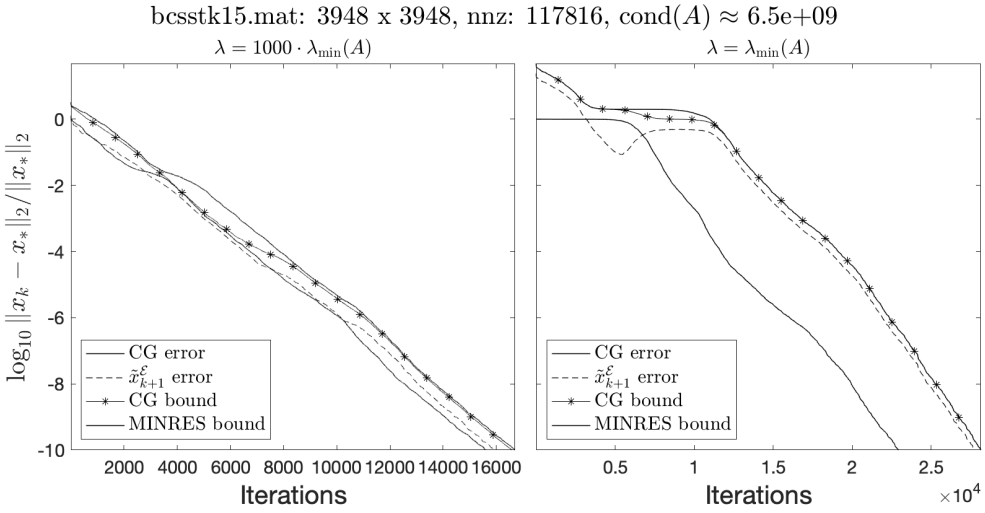

FIG. 7.3. *Different regularization parameters for an SPD problem. Left: Our error estimate is tighter when λ is large compared to $\lambda_{\min}(A)$. Right: Our error estimate quickly converges to the MINRES-based bound.*

following the method of [17, Alg. 5] and costing $\mathcal{O}(1)$ per iteration. Our observations mirror those made in [17]: better approximations $\tilde{\sigma}$ lead to tighter error bounds, but beyond a certain point the bounds become the same as the robust LSMR-based bound.

When we use $\sigma_{\min}(R_k)$ as a proxy for $\sigma_{\min}(A)$ the robust LSMR-based estimate is not monotonically decreasing (since $\sigma_{\min}(R_k)$ is decreasing), nor is it guaranteed to be an upper bound. Given sufficient time for the smallest singular value to converge, it may nonetheless be useful for practical purposes—see [17] for a more thorough discussion of methods for estimating $\sigma_{\min}(R_k)$.

In a third set of experiments (see Figure 7.3), we treated $\sigma_{\min}(A)$ as unknown but used varying regularization parameters λ. In general, using larger values for λ had an

effect on our estimates similar to using tighter values for $\tilde{\sigma}$. Our error estimates were smaller than the corresponding LSMR-based estimates, though never by much, and these two estimates converged as the error decreased.

We also ran these experiments on real SPD matrices from the SuiteSparse collection, solving $Ax = b$ or $(A + \lambda I)x = b$ with $b = [1, 1, \ldots, 1]/\sqrt{n}$. Our observations were essentially the same, except that since there was no equivalent to Craig's method we observed no "first phase" in which the errors for $\tilde{x}_{k+1}$ or $\tilde{x}^{\mathcal{E}}_{k+1}$ would increase. The errors for these points would still increase later on, around the time when the quality of our error estimates degraded.

Given the results of these experiments, it is not entirely clear whether it is better to switch to $\tilde{x}^{(G)}_{k+1}$ or $\tilde{x}^{\mathcal{E}(G)}_{k+1}$ upon termination or to continue to use x^Q_k. If a user adopts the conservative stance that an iterative method is only as good as its best guaranteed error bound, then $\tilde{x}^{\mathcal{E}(G)}_{k+1}$ is always superior to x^Q_k. However, our experiments suggest that whenever the relative error is below 10^{-8} or so, x^Q_k is highly likely to have the smaller true error. In many situations it may therefore be preferable simply to use the LSQR or CG iterates.

8. Conclusions. Given a lower bound $\tilde{\sigma}$ on $\sigma_{\min}(A)$ or a regularization parameter $\lambda > 0$, we have derived estimates for the LSQR error $\|x^Q_k - x_*\|_2$ and similar estimates for the CG error. We have also shown how to find a point that triggers our stopping critera sooner than LSQR or CG would, although its true error is often larger than the LSQR/CG error. All of the necessary computation can be done for only $\mathcal{O}(1)$ work beyond that already required by LSQR, since we can cheaply track the error estimates and transfer from LSQR to the desired point only upon termination.

We reiterate two caveats: first, we have assumed exact arithmetic throughout this paper, and so our improved bounds may not necessarily hold in finite precision. Second, in practice the LSQR point x^Q_k will often have a smaller error than the point $\tilde{x}^{\mathcal{E}(G)}_{k+1}$ that minimizes our error bound. Taking these two facts into consideration, as well as the fact that our new error estimates are at most a factor of 2 smaller than existing estimates that hold in finite precision, we recommend using LSQR in practice.

The more significant contribution of this paper is the tightness result of Theorem 3.8. Our bounds are the tightest estimates possible if we only use the information gained from running the Golub–Kahan (resp., Lanczos) process, plus $\tilde{\sigma}$ and λ. Thus future work in developing practical stopping rules should either use additional information about A and b or focus on developing error estimates that are not guaranteed upper bounds. The latter option will be especially practical in situations where $\sigma_{\min}(A)$ is not known ahead of time.

MATLAB implementations for both CG and LSQR are available at https://erhallma.math.ncsu.edu/forward/.

Acknowledgments. The author would like to thank the referees for their detailed comments, which have greatly improved the presentation of this paper. Thanks also go to Jonathan Leake, whose comments helped to improve the exposition.

REFERENCES

[1] T. Davis and Y. Hu, *The University of Florida Sparse Matrix Collection*, ACM Trans. Math. Software, 38 (2011), pp. 1:1–1:25.

[2] R. Estrin, D. Orban, and M. Saunders, *Euclidean-norm error bounds for SYMMLQ and CG*, SIAM J. Matrix Anal. Appl., 40 (2019), pp. 235–253.

[3] R. Estrin, D. Orban, and M. Saunders, *LNLQ: An iterative method for least-norm problems with an error minimization property*, SIAM J. Matrix Anal. Appl., 40 (2019), pp. 1102–1124.
[4] R. Estrin, D. Orban, and M. Saunders, *LSLQ: An iterative method for linear least-squares with an error minimization property*, SIAM J. Matrix Anal. Appl., 40 (2019), pp. 254–275.
[5] D. Fadeev and V. Fadeeva, *Computational Methods of Linear Algebra*, Freeman, San Francisco, 1963.
[6] D. Fong and M. Saunders, *LSMR: An iterative algorithm for sparse least squares problems*, SIAM J. Sci. Comput., 33 (2011), pp. 2950–2971.
[7] A. Frommer, K. Kahl, T. Lippert, and H. Rittich, 2*-norm error bounds and estimates for Lanczos approximations to linear systems and rational matrix functions*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 1046–1065.
[8] G. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, SIAM J. Numer. Anal., 2 (1965), pp. 205–224.
[9] G. Golub and G. Meurant, *Matrices, moments and quadrature* II*: How to compute the norm of the error in iterative methods*, BIT, 37 (1997), pp. 687–705.
[10] E. Hallman, *Error Estimates for Least-Squares Problems*, Ph.D. thesis, University of California, Berkeley, 2019.
[11] E. Hallman and M. Gu, *LSMB: Minimizing the backward error for least-squares problems*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 1295–1317.
[12] M. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Natl. Bur. Stand., 49 (1952), pp. 409–436.
[13] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Natl. Bur. Stand., 45 (1950), pp. 255–282.
[14] G. Meurant, *The computation of bounds for the norm of the error in the conjugate gradient algorithm*, Numer. Algorithms, 16 (1997), pp. 77–87.
[15] G. Meurant, *Estimates of the l2 norm of the error in the conjugate gradient algorithm*, Numer. Algorithms, 40 (2005), pp. 157–169.
[16] G. Meurant and P. Tichý, *On computing quadrature-based bounds for the A-norm of the error in conjugate gradients*, Numer. Algorithms, 62 (2013), pp. 163–191.
[17] G. Meurant and P. Tichý, *Approximating the extreme Ritz values and upper bounds for the A-norm of the error in CG*, Numer. Algorithms, 82 (2019), pp. 937–968.
[18] C. Paige, *Bidiagonalization of matrices and solution of linear equations*, SIAM J. Numer. Anal., 11 (1974), pp. 197–209.
[19] C. Paige and M. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.
[20] C. Paige and M. Saunders, *A Bidiagonalization Algorithm for Sparse Linear Equations and Least-Squares Problems*, Tech. Report SOL-78-19, Stanford University, Stanford, CA, 1978.
[21] C. Paige and M. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 43–71.
[22] M. Saunders, *Solution of sparse rectangular systems using LSQR and CRAIG*, BIT, 35 (1995), pp. 588–604.
[23] P. Tichý, *A New Algorithm for Computing Quadrature-Based Bounds in Conjugate Gradients*, http://www.cs.cas.cz/tichy/download/present/2014Spa.pdf, 2014.