LNLQ: AN ITERATIVE METHOD FOR LEAST-NORM PROBLEMS WITH AN ERROR MINIMIZATION PROPERTY

RON ESTRIN*, DOMINIQUE ORBAN†, AND MICHAEL A. SAUNDERS‡

Abstract. We describe LNLQ for solving the least-norm problem min $\|x\|$ subject to $Ax = b$, using the Golub-Kahan bidiagonalization of $\begin{bmatrix} b & A \end{bmatrix}$. Craig's method is known to be equivalent to applying the conjugate gradient method to the normal equations of the second kind ($AA^T y = b$, $x = A^T y$); LNLQ is equivalent to applying SYMMLQ. If an underestimate of the smallest singular value is available, error upper bounds for both x and y are available cheaply at each iteration. LNLQ is a companion method to the least-squares solver LSLQ (Estrin, Orban, and Saunders, 2019b), which is equivalent to SYMMLQ on the conventional normal equations. We show that the error bounds are tight and comparable to the bounds suggested by Arioli (2013) for CRAIG. A sliding window technique allows us to tighten the error bound for y at the expense of a few additional scalar operations per iteration. We illustrate the tightness of the error bounds on two standard test problems and on the computation of an inexact gradient in the context of a penalty method for PDE-constrained optimization.

Key words. Linear least-norm problem, error minimization, SYMMLQ, CG, CRAIG.

AMS subject classifications. 15A06, 65F10, 65F22, 65F25, 65F35, 65F50, 93E24

1. Introduction.

We seek the unique $x_\star$ that solves the least-norm problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \ \tfrac{1}{2}\|x\|^2 \ \text{ subject to } \ Ax = b, \tag{1}$$

where $\|\cdot\|$ denotes the Euclidean norm, $A \in \mathbb{R}^{m \times n}$, and the constraints are assumed to be consistent. A unique $y_\star$ solves the problem

$$\underset{y \in \mathbb{R}^m}{\text{minimize}} \ \tfrac{1}{2}\|y\|^2 \ \text{ subject to } \ AA^T y = b, \tag{2}$$

and $(x_\star, y_\star)$ is the least-norm solution of the normal equations of the second kind:

$$AA^T y = b, \quad x = A^T y \qquad \Leftrightarrow \qquad \begin{bmatrix} -I & A^T \\ A & \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix}. \tag{3}$$

We describe an iterative solver LNLQ that includes cheap and reliable upper bounds on the sequence of errors $\|x_k - x_\star\|$ and $\|y_k - y_\star\|$.

Existing iterative methods tailored to the solution of (1) include CRAIG (Craig, 1955) and LSQR (Paige and Saunders, 1982a,b). LSQR does not provide convenient error bounds. CRAIG generates iterates x_k that are updated along orthogonal directions, so it is possible to devise an upper bound on the error in x_k (Arioli, 2013), but the iterates y_k are not updated along orthogonal directions.

*Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA, USA. E-mail: restrin@stanford.edu.

†GERAD and Department of Mathematics and Industrial Engineering, École Polytechnique, Montréal, QC, Canada. E-mail: dominique.orban@gerad.ca. Research partially supported by an NSERC Discovery Grant.

‡Systems Optimization Laboratory, Department of Management Science and Engineering, Stanford University, Stanford, CA, USA. E-mail: saunders@stanford.edu. Research partially supported by the National Institute of General Medical Sciences of the National Institutes of Health, award U01GM102098. Version of July 7, 2018.

CRAIG and LSQR turn out to be formally equivalent to the method of conjugate gradients (CG) (Hestenes and Stiefel, 1952) and MINRES (Paige and Saunders, 1975) applied to $AA^Ty = b$ in (3), respectively, but are more reliable when A is ill-conditioned. By construction, LNLQ is formally equivalent to SYMMLQ applied to (3). LNLQ inherits beneficial properties of SYMMLQ, including orthogonal updates to y_k, cheap transfers to the CRAIG point, and cheap upper bounds on the error $\|y_k - y_\star\|$.

Motivation. Linear systems of the form (3) occur during evaluation of the value and gradient of a certain penalty function for equality-constrained optimization (Fletcher, 1973; Estrin, Friedlander, Orban, and Saunders, 2018). Our main motivation is to devise reliable termination criteria that allow control of the error in the solution of (1), thus allowing us to evaluate inexact gradients cheaply while maintaining global convergence properties of the underlying optimization method. Our approach follows the philosophy of Estrin, Orban, and Saunders (2019a) and Estrin et al. (2019b) and requires an estimate of the smallest singular value of A. Although such an estimate may not always be available in practice, good underestimates are often available in optimization problems, including PDE-constrained problems—see section 7.

Arioli (2013) develops an upper bound on the error in x_k along the CRAIG iterations based on an appropriate Gauss-Radau quadrature (Golub and Meurant, 1994), and suggests the seemingly simplistic upper bound $\|y_k - y_\star\| \leqslant \|x_k - x_\star\|/\sigma_r$, where σ_r is the smallest nonzero singular value of A. Although his bound is often effective, we derive improved bounds for CRAIG using LNLQ by introducing a delay d as in (Golub and Strakŏs, 1994).

The remainder of this paper is outlined as follows. Section 2 gives background on the Golub and Kahan (1965) process and CRAIG. Sections 3–6 derive LNLQ from the Golub and Kahan process, highlight relationships to CRAIG, derive error bounds, and discuss regularization and preconditioning. Numerical experiments are given in section 7. Extensions to quasi-definite systems are given in section 8, followed by concluding remarks in section 9.

Notation and assumptions. We use Householder notation: A, b, β for matrix, vector, scalar, with the exception of c and s denoting scalars that define reflections. All vectors are columns, but the slightly abusive notation $(\xi_1, \dots, \xi_k)$ is sometimes used to enumerate their components in the text. Unless specified otherwise, $\|A\|$ and $\|x\|$ denote the Euclidean norm of matrix A and vector x. For symmetric positive definite M, we define the M-norm of u via $\|u\|_M^2 := u^TMu$. We order the singular values of A according to $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_{\min(m,n)} \geqslant 0$, and $A^\dagger$ denotes the Moore-Penrose pseudoinverse of A. We assume that $x_0 = 0$ and $y_0 = 0$. If $y_0 \neq 0$, we can solve the shifted system $AA^T\Delta y = b - AA^Ty_0$ and set $y = y_0 + \Delta y$.

As in Estrin et al. (2019a), in the derivation of some results we rely on orthogonality of the columns of the Golub-Kahan matrices U_k, V_k. In practice, the orthogonality is lost and the convergence of our method is delayed. Nevertheless, the method as well as the error upper bounds derived using the orthogonality assumption remain reliable, as observed empirically. Analysis of this phenomenon is beyond the scope of this paper.

2. Background.

2.1. The Golub-Kahan process.

The Golub and Kahan (1965) process applied to A with starting vector b is described as Algorithm 1. In line 1, $\beta_1 u_1 = b$ is short for "$\beta_1 = \|b\|$; if $\beta_1 = 0$ then exit; else $u_1 = b/\beta_1$". Similarly for line 2 and the main loop. In exact arithmetic, the algorithm terminates with $k = \ell \leqslant \min(m, n)$ and either $\alpha_{\ell+1}$ or $\beta_{\ell+1} = 0$. Paige (1974) explains that if $Ax = b$ is consistent, the process must

Algorithm 1 Golub-Kahan Bidiagonalization Process

Require: A, b

$\beta_1 u_1 = b$
$\alpha_1 v_1 = A^T u_1$
for $k = 1, 2, \dots$ **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

terminate with $\beta_{\ell+1} = 0$.

We define $U_k := \begin{bmatrix} u_1 & \cdots & u_k \end{bmatrix}$, $V_k := \begin{bmatrix} v_1 & \cdots & v_k \end{bmatrix}$, and

$$L_k := \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \end{bmatrix}, \quad B_k := \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{bmatrix} = \begin{bmatrix} L_k \\ \beta_{k+1} e_k^T \end{bmatrix}. \tag{4}$$

After k iterations of Algorithm 1, the following hold to machine precision:

$$A V_k = U_{k+1} B_k, \tag{5a}$$

$$A^T U_{k+1} = V_k B_k^T + \alpha_{k+1} v_{k+1} e_{k+1}^T = V_{k+1} L_{k+1}^T, \tag{5b}$$

while the identities $U_k^T U_k = I_k$ and $V_k^T V_k = I_k$ hold only in exact arithmetic. The next sections assume that these identities do hold, allowing us to derive certain norm estimates that seem reliable in practice until high accuracy is achieved in x and y.

2.2. CRAIG. For problem (1), the method of Craig (1955) was originally derived as a form of the conjugate gradient (CG) method (Hestenes and Stiefel, 1952) applied to (3). Paige (1974) provided a description based on Algorithm 1:

$$L_k t_k = \beta_1 e_1, \qquad x_k^C := V_k t_k = x_{k-1}^C + \tau_k v_k, \tag{6}$$

where $t_k := (\tau_1, \dots, \tau_k)$ and the components of t_k can be found recursively from $\tau_1 = \beta_1/\alpha_1$, $\tau_j = -\beta_j \tau_{j-1}/\alpha_j$ $(j \geqslant 2)$. If we suppose $t_k = L_k^T \bar{y}_k^C$ for some vector $\bar{y}_k^C$ that exists but need not be computed, we see that

$$x_k^C = V_k L_k^T \bar{y}_k^C = A^T U_k \bar{y}_k^C = A^T y_k^C, \tag{7}$$

where $y_k^C := U_k \bar{y}_k^C$ provides approximations to y. If we define $D_k = \begin{bmatrix} d_1 & \cdots & d_k \end{bmatrix}$ from $L_k D_k^T = U_k^T$, we may compute the vectors d_j recursively from $d_1 = u_1/\alpha_1$, $d_j = u_j - \beta_j d_{j-1}/\alpha_j$ $(j \geqslant 2)$ and then update

$$y_k^C = D_k L_k^T \bar{y}_k^C = D_k t_k = y_{k-1}^C + \tau_k d_k.$$

To see the equivalence with CG on (3), note that relations (5) yield

$$A A^T U_k = A V_k L_k^T = U_{k+1} B_k L_k^T = U_{k+1} H_k, \tag{8}$$

$$H_k := B_k L_k^T = \begin{bmatrix} L_k L_k^T \\ \alpha_k \beta_{k+1} e_k^T \end{bmatrix}, \tag{9}$$

which we recognize as the result of k iterations of the Lanczos (1950) process applied to AA^T with starting vector b, where

$$T_k := L_k L_k^T = \begin{bmatrix} \bar{\alpha}_1 & \bar{\beta}_2 & & \\ \bar{\beta}_2 & \bar{\alpha}_2 & \ddots & \\ & \ddots & \ddots & \bar{\beta}_k \\ & & \bar{\beta}_k & \bar{\alpha}_k \end{bmatrix} \tag{10}$$

is the Cholesky factorization of the Lanczos tridiagonal T_k, with $\bar{\alpha}_1 := \alpha_1^2$ and $\bar{\alpha}_j := \alpha_j^2 + \beta_j^2$, $\bar{\beta}_j := \alpha_j \beta_{j+1}$ for $j \geqslant 2$. Note that $T_k \bar{y}_k^C = L_k L_k^T \bar{y}_k^C = L_k t_k = \beta_1 e_1$. CG defines $y_k^C = U_k \bar{y}_k^C$, and so we have the same iterates as CRAIG:

$$x_k^C = A^T y_k^C = A^T U_k \bar{y}_k^C = V_k L_k^T \bar{y}_k^C = V_k t_k = x_{k-1}^C + \tau_k v_k.$$

While D_k is not orthogonal, note that x_k^C in (6) is updated along orthogonal directions and $\|x_k^C\|^2 = \sum_{j=1}^k \tau_j^2$, i.e., $\|x_k^C\|$ is monotonically increasing and $\|x_\star - x_k^C\|$ is monotonically decreasing. Arioli (2013) exploits these facts to compute upper and lower bounds on the error $\|x_\star - x_k^C\|$ and an upper bound on $\|y_\star - y_k^C\|$.

Although it is not apparent in the above derivation, the equivalence with CG applied to (3) shows that $\|y_k^C\|$ is monotonically increasing and $\|y_\star - y_k^C\|$ is monotonically decreasing (Hestenes and Stiefel, 1952, Theorem 6:3).

Unfortunately, the fact that y_k^C is not updated along orthogonal directions makes it more difficult to monitor $\|y_\star - y_k^C\|$ and to develop upper and lower bounds. Arioli (2013) suggests the upper bound $\|y_\star - y_k^C\| \leqslant \|x_\star - x_k^C\| / \sigma_n$ when A has full row rank. LNLQ provides an alternative upper bound on $\|y_\star - y_k^C\|$ that may be tighter.

The residual for CRAIG is

$$r_k^C := b - A x_k^C = \beta_1 u_1 - A V_k t_k = U_{k+1}(\beta_1 e_1 - B_k t_k) = -\beta_{k+1} \tau_k u_{k+1}. \tag{11}$$

Other results may be found scattered in the literature. For completeness, we gather them here and provide proofs.

PROPOSITION 1. *Let $x_\star$ be the solution of (1) and $y_\star$ the associated Lagrange multiplier with minimum norm, i.e., the minimum-norm solution of (3). The kth CRAIG iterates x_k^C and y_k^C solve*

$$\underset{x}{\text{minimize}} \ \|x - x_\star\| \ \text{ subject to } \ x \in \text{Range}(V_k), \tag{12}$$

$$\underset{y}{\text{minimize}} \ \|y - y_\star\|_{AA^T} \ \text{ subject to } \ y \in \text{Range}(U_k) \tag{13}$$

respectively. In addition, x_k^C and y_k^C solve

$$\underset{x}{\text{minimize}} \ \|x\| \ \text{ subject to } \ x \in \text{Range}(V_k),\ b - Ax \perp \text{Range}(U_k), \tag{14}$$

$$\underset{y}{\text{minimize}} \ \|y\|_{AA^T} \ \text{ subject to } \ y \in \text{Range}(U_k),\ b - AA^T y \perp \text{Range}(U_k). \tag{15}$$

When A is row-rank-deficient, the (AA^T)-norm should be interpreted as a norm when restricted to Range(A).

Proof. Assume temporarily that A has full row rank, so that AA^T is symmetric positive definite. Then there exists a unique $y_\star$ such that $x_\star = A^T y_\star$ and

$$\|x_k^C - x_\star\| = \|A^T(y_k^C - y_\star)\| = \|y_k^C - y_\star\|_{AA^T}.$$

In words, the Euclidean norm of the error in x_k is the energy norm of the error in y_k. Theorem 6:1 of Hestenes and Stiefel (1952) ensures that y_k^C is chosen to minimize the energy norm of the error over all $y \in \operatorname{Range}(U_k)$, i.e., y_k^C solves (13).

To $y \in \operatorname{Range}(U_k)$, there corresponds $x = A^T y \in \operatorname{Range}(A^T U_k) = \operatorname{Range}(V_k L_k^T) = \operatorname{Range}(V_k)$ by (5) because L_k is nonsingular. Consequently, CRAIG generates x_k^C as a solution of (12).

When A is rank-deficient, our assumption that $Ax = b$ is consistent ensures that $AA^T y = b$ is also consistent because if there exists a subpace of solutions x, it is possible to pick the one that solves (3), and therefore $b \in \operatorname{Range}(AA^T)$. Kammerer and Nashed (1972) show that in the consistent singular case, CG converges to the solution $y_\star$ of (2). Let $r < \min(m, n)$ be such that $\sigma_r > 0$ and $\sigma_{r+1} = \cdots = \sigma_{\min(m,n)} = 0$. Then $\operatorname{rank}(A) = r = \dim \operatorname{Range}(A)$ and the smallest nonzero eigenvalue of AA^T is σ_r^2. The Rayleigh-Ritz theorem states that

$$\sigma_r^2 = \min \{\|A^T w\|^2 \mid w \in \operatorname{Range}(A),\ \|w\| = 1\}.$$

By (5), each $u_k \in \operatorname{Range}(A)$, and (8) and (10) imply that $U_k^T AA^T U_k = T_k$ in exact arithmetic. Thus for any $t \in \mathbb{R}^k$ such that $\|t\| = 1$, we have $\|U_k t\| = 1$ and

$$t^T U_k^T AA^T U_k t = t^T T_k t \geqslant \sigma_r^2,$$

so that the T_k are uniformly positive definite and CG iterations occur as if CG were applied to the positive-definite reduced system $P_r^T AA^T P_r \tilde{y} = P_r^T b$, where P_r is the $m \times r$ matrix of orthogonal eigenvectors of AA^T corresponding to nonzero eigenvalues. Thus in the rank-deficient case, y_k^C also solves (13) except that the energy "norm" is only a norm when restricted to $\operatorname{Range}(A)$, and x_k^C also solves (12).

To establish (14), note that (6) and (11) imply x_k^C is primal feasible for (14). Dual feasibility requires there exist vectors $\bar{x}$, $\bar{y}$ and $\bar{z}$ such that $x = \bar{z} + A^T U_k \bar{y}$, $V_k^T \bar{z} = 0$ and $x = V_k \bar{x}$. The first two conditions are equivalent to $V_k^T x = 0 + V_k^T A^T U_k \bar{y} = B_k^T U_{k+1}^T U_k \bar{y} = L_k^T \bar{y}$. Because $x = V_k \bar{x}$, this amounts to $\bar{x} = L_k^T \bar{y}$. Thus dual feasibility is satisfied with $\bar{x} := \bar{x}_k^C$, $\bar{y} := \bar{y}_k^C$ and $\bar{z} := 0$. The proof of (15) is similar. ☐

3. LNLQ.

We define LNLQ as equivalent in exact arithmetic to SYMMLQ (Paige and Saunders, 1975) applied to (3). Whereas SYMMLQ is based on the Lanczos (1950) process, LNLQ is based on Algorithm 1. Again we seek an approximation $y_k^L = U_k \bar{y}_k^L$. The kth iteration of SYMMLQ applied to (3) computes $\bar{y}_k^L$ as the solution of

$$\underset{\bar{y}}{\text{minimize}} \ \ \tfrac{1}{2}\|\bar{y}\|^2 \quad \text{subject to} \ \ H_{k-1}^T \bar{y} = \beta_1 e_1, \tag{16}$$

where H_{k-1}^T is the top $(k-1) \times k$ submatrix of T_k (10).

3.1. An LQ factorization.

In SYMMLQ, the computation of $\bar{y}_k^L$ follows from the LQ factorization of H_{k-1}^T, which can be derived implicitly via the LQ factorization of $T_k = L_k L_k^T$. As L_k is already lower triangular, we only need the factorization

$$L_k^T = \overline{M}_k Q_k, \qquad \overline{M}_k := \begin{bmatrix} \varepsilon_1 & & & \\ \eta_2 & \varepsilon_2 & & \\ & \ddots & \ddots & \\ & & \eta_k & \bar{\varepsilon}_k \end{bmatrix} = \begin{bmatrix} M_{k-1} & \\ \eta_k e_{k-1}^T & \bar{\varepsilon}_k \end{bmatrix}, \tag{17}$$

where $Q_k^T = Q_{1,2}Q_{2,3}\dots Q_{k-1,k}$ is orthogonal and defined as a product of reflections, where $Q_{j-1,j}$ is the identity except for elements at the intersection of rows and columns $j-1$ and j. Initially, $\bar\varepsilon_1 = \alpha_1$ and $Q_1 = I$. Subsequent factorization steps may be represented as

$$\begin{array}{c c} & \begin{array}{ccc} \scriptstyle j-2 & \scriptstyle j-1 & \scriptstyle j \end{array} \\ \begin{array}{c} \scriptstyle j-1 \\ \scriptstyle j \end{array} & \begin{bmatrix} \eta_{j-1} & \bar\varepsilon_{j-1} & \beta_j \\ & & \alpha_j \end{bmatrix} \end{array} \begin{array}{c} \begin{array}{ccc} \scriptstyle j-2 & \scriptstyle j-1 & \scriptstyle j \end{array} \\ \begin{bmatrix} 1 & & \\ & c_j & s_j \\ & s_j & -c_j \end{bmatrix} \end{array} = \begin{array}{c} \begin{array}{ccc} \scriptstyle j-2 & \scriptstyle j-1 & \scriptstyle j \end{array} \\ \begin{bmatrix} \eta_{j-1} & \varepsilon_{j-1} & \\ & \eta_j & \bar\varepsilon_j \end{bmatrix} \end{array},$$

where the border indices indicate row and column numbers, with the understanding that η_{j-1} is absent when $j = 2$. For $j \geqslant 2$, $Q_{j-1,j}$ is defined by

$$\varepsilon_{j-1} = \sqrt{\bar\varepsilon_{j-1}^2 + \beta_j^2}, \quad c_j = \bar\varepsilon_{j-1}/\varepsilon_{j-1}, \quad s_j = \beta_j/\varepsilon_{j-1},$$

and the application of $Q_{j-1,j}$ results in

$$\eta_j = \alpha_j s_j, \quad \bar\varepsilon_j = -\alpha_j c_j. \tag{18}$$

We may write $H_{k-1}^T = \begin{bmatrix} L_{k-1}L_{k-1}^T & \alpha_{k-1}\beta_k e_{k-1} \end{bmatrix} = L_{k-1}\begin{bmatrix} L_{k-1}^T & \beta_k e_{k-1} \end{bmatrix}$. From (17),

$$L_k^T = \begin{bmatrix} L_{k-1}^T & \beta_k e_{k-1} \\ & \alpha_k \end{bmatrix} = \begin{bmatrix} M_{k-1} & \\ \eta_k e_{k-1}^T & \bar\varepsilon_k \end{bmatrix} Q_k \quad \Rightarrow \quad \begin{bmatrix} L_{k-1}^T & \beta_k e_{k-1} \end{bmatrix} = \begin{bmatrix} M_{k-1} & 0 \end{bmatrix} Q_k.$$

Finally, we obtain the LQ factorization

$$H_{k-1}^T = \begin{bmatrix} L_{k-1}M_{k-1} & 0 \end{bmatrix} Q_k. \tag{19}$$

3.2. Definition and update of the LNLQ and CRAIG iterates. In order to solve $H_{k-1}^T \bar y_k^L = \beta_1 e_1$ using (19), we already have $L_{k-1}t_{k-1} = \beta_1 e_1$, with the next iteration giving $\tau_k = -\beta_k \tau_{k-1}/\alpha_k$. Next, we consider $M_{k-1}z_{k-1} = t_{k-1}$ and find the components of $z_{k-1} = (\zeta_1, \dots, \zeta_{k-1})$ recursively as $\zeta_1 = \tau_1/\varepsilon_1$, $\zeta_j = (\tau_j - \eta_j\zeta_{j-1})/\varepsilon_j$ $(j \geqslant 2)$. This time, the next iteration yields $\bar\zeta_k = (\tau_k - \eta_k\zeta_{k-1})/\bar\varepsilon_k$ and $\zeta_k = \bar\zeta_k\bar\varepsilon_k/\varepsilon_k = c_{k+1}\bar\zeta_k$. Thus,

$$\bar y_k^L = Q_k^T \begin{bmatrix} z_{k-1} \\ 0 \end{bmatrix} \qquad \text{and} \qquad \bar y_k^C = Q_k^T \begin{bmatrix} z_{k-1} \\ \bar\zeta_k \end{bmatrix} = Q_k^T \bar z_k \tag{20}$$

solve (16) and $T_k \bar y_k^C = \beta_1 e_1$ respectively, matching the definition of the CRAIG iterate.

By construction, $y_k^L = U_k \bar y_k^L$ and $y_k^C = U_k \bar y_k^C$. We define the orthogonal matrix

$$\overline{W}_k = U_k Q_k^T = \begin{bmatrix} w_1 & \cdots & w_{k-1} & \bar w_k \end{bmatrix} = \begin{bmatrix} W_{k-1} & \bar w_k \end{bmatrix}, \quad \bar w_1 := u_1,$$

so that (20) with z_{k-1} and $\bar z_k := (z_{k-1}, \bar\zeta_k)$ yields the orthogonal updates

$$y_k^L = \overline{W}_k \begin{bmatrix} z_{k-1} \\ 0 \end{bmatrix} = W_{k-1}z_{k-1} = y_{k-1}^L + \zeta_{k-1}w_{k-1}, \tag{21}$$

$$y_k^C = \overline{W}_k \bar z_k = W_{k-1}z_{k-1} + \bar\zeta_k \bar w_k = y_k^L + \bar\zeta_k \bar w_k. \tag{22}$$

Because $\overline{W}_k$ is orthogonal, we have

$$\|y_k^L\|^2 = \|z_{k-1}\|^2 = \sum_{j=1}^{k-1} \zeta_j^2 \quad \text{and} \quad \|y_k^C\|^2 = \|y_k^L\|^2 + \bar\zeta_k^2. \tag{23}$$

Thus $\|y_k^C\| \geqslant \|y_k^L\|$, $\|y_k^L\|$ is monotonically increasing, $\|y_\star - y_k^L\|$ is monotonically decreasing, and $\|y_\star - y_k^L\| \geqslant \|y_\star - y_k^C\|$, consistent with (Estrin et al., 2019a, Theorem 6).

Contrary to the update of y_k^C in CRAIG, y_k^L is updated along orthogonal directions and y_k^C is found as an orthogonal update of y_k^L. The latter follows from the transfer procedure of SYMMLQ to the CG point described by Paige and Saunders (1975).

At the next iteration,

$$
\begin{aligned}
\begin{bmatrix} w_k & \bar{w}_{k+1} \end{bmatrix} &= \begin{bmatrix} \bar{w}_k & u_{k+1} \end{bmatrix} \begin{array}{c} \begin{array}{cc} \scriptstyle k & \scriptstyle k+1 \end{array} \\ \begin{bmatrix} c_{k+1} & s_{k+1} \\ s_{k+1} & -c_{k+1} \end{bmatrix} \end{array} \\
\Rightarrow \quad w_k &= c_{k+1}\bar{w}_k + s_{k+1}u_{k+1}, \\
\bar{w}_{k+1} &= s_{k+1}\bar{w}_k - c_{k+1}u_{k+1}.
\end{aligned}
$$

3.3. Residual estimates. We define the residual

$$
r_k := b - Ax_k = b - AA^T U_k \bar{y}_k = U_{k+1}(\beta_1 e_1 - H_k \bar{y}_k)
$$

using line 1 of Algorithm 1 and (8), where $\bar{y}_k$ is either $\bar{y}_k^L$ or $\bar{y}_k^C$. Then for $k > 1$,

$$
\begin{aligned}
T_k \bar{y}_k^L = L_k L_k^T \bar{y}_k^L &= L_k \overline{M}_k Q_k Q_k^T \begin{bmatrix} z_{k-1} \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} L_{k-1} & \\ \beta_k e_{k-1}^T & \alpha_k \end{bmatrix} \begin{bmatrix} M_{k-1} & \\ \eta_k e_{k-1}^T & \bar{\varepsilon}_k \end{bmatrix} \begin{bmatrix} z_{k-1} \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} L_{k-1} & \\ \beta_k e_{k-1}^T & \alpha_k \end{bmatrix} \begin{bmatrix} t_{k-1} \\ \eta_k \zeta_{k-1} \end{bmatrix} = \begin{bmatrix} \beta_1 e_1 \\ \beta_k \tau_{k-1} + \alpha_k \eta_k \zeta_{k-1} \end{bmatrix},
\end{aligned}
$$

where we use (17), the definition of t_{k-1} and z_{k-1}, and (20). Also, the identity $Q_k e_k = s_k e_{k-1} - c_k e_k$ yields

$$
e_k^T \bar{y}_k^L = e_k^T Q_k^T \begin{bmatrix} z_{k-1} \\ 0 \end{bmatrix} = s_k \zeta_{k-1}.
$$

These combine with (9) to give

$$
\begin{aligned}
r_k^L &= U_{k+1} \left(\begin{bmatrix} \beta_1 e_1 \\ 0 \end{bmatrix} - \begin{bmatrix} L_k L_k^T \\ \bar{\beta}_{k+1} e_k^T \end{bmatrix} \bar{y}_k^L \right) = -U_{k+1} \begin{bmatrix} 0 \\ \beta_k \tau_{k-1} + \alpha_k \eta_k \zeta_{k-1} \\ \bar{\beta}_{k+1} s_k \zeta_{k-1} \end{bmatrix} \\
&= -(\beta_k \tau_{k-1} + \alpha_k \eta_k \zeta_{k-1}) u_k - \bar{\beta}_{k+1} s_k \zeta_{k-1} u_{k+1}. \qquad (24)
\end{aligned}
$$

By orthogonality, the residual norm is cheaply computable as

$$
\|r_k^L\|^2 = (\beta_k \tau_{k-1} + \alpha_k \eta_k \zeta_{k-1})^2 + (\bar{\beta}_{k+1} s_k \zeta_{k-1})^2.
$$

Similarly,

$$
\begin{aligned}
r_k^C &= U_{k+1} \left(\begin{bmatrix} \beta_1 e_1 \\ 0 \end{bmatrix} - \begin{bmatrix} T_k \\ \bar{\beta}_{k+1} e_k^T \end{bmatrix} \bar{y}_k^C \right) = -U_{k+1} \begin{bmatrix} 0 \\ \bar{\beta}_{k+1} e_k^T \end{bmatrix} Q_k^T \bar{z}_k \\
&= -\bar{\beta}_{k+1} U_{k+1} \begin{bmatrix} 0 \\ s_k e_{k-1}^T - c_k e_k^T \end{bmatrix} \begin{bmatrix} z_{k-1} \\ \bar{\zeta}_k \end{bmatrix} \\
&= -\bar{\beta}_{k+1}(s_k \zeta_{k-1} - c_k \bar{\zeta}_k) u_{k+1}, \qquad (25)
\end{aligned}
$$

where we use $T_k \bar{y}_k^C = \beta_1 e_1$ (by definition) and (20). Orthogonality of the u_j yields orthogonality of the CRAIG residuals, a property of CG (Hestenes and Stiefel, 1952, Theorem 5:1). The CRAIG residual norm is simply

$$\|r_k^C\| = \bar{\beta}_{k+1} \, |s_k \zeta_{k-1} - c_k \bar{\zeta}_k|.$$

In the next section, alternative expressions of $\|r_k^L\|$ and $\|r_k^C\|$ emerge.

3.4. Updating $\boldsymbol{x = A^T y}$. The definition $y_k = U_k \bar{y}_k$ and (5) yield $x_k = A^T y_k = A^T U_k \bar{y}_k = V_k L_k^T \bar{y}_k$. The LQ and CRAIG iterates may then be updated as

$$\begin{aligned} x_k^L &= V_k L_k^T \bar{y}_k^L = V_k L_k^T Q_k \begin{bmatrix} z_{k-1} \\ 0 \end{bmatrix} \\ &= V_k \overline{M}_k \begin{bmatrix} z_{k-1} \\ 0 \end{bmatrix} = V_k \begin{bmatrix} M_{k-1} & \\ \eta_k e_{k-1}^T & \bar{\varepsilon}_k \end{bmatrix} \begin{bmatrix} z_{k-1} \\ 0 \end{bmatrix} \\ &= V_{k-1} M_{k-1} z_{k-1} + \eta_k \zeta_{k-1} v_k \\ &= V_{k-1} t_{k-1} + \eta_k \zeta_{k-1} v_k, \end{aligned} \tag{26}$$

and similarly,

$$x_k^C = V_k \begin{bmatrix} M_{k-1} & \\ \eta_k e_{k-1}^T & \bar{\varepsilon}_k \end{bmatrix} \begin{bmatrix} z_{k-1} \\ \bar{\zeta}_k \end{bmatrix} = x_k^L + \bar{\varepsilon}_k \bar{\zeta}_k v_k. \tag{27}$$

Because V_k is orthogonal, we have

$$\|x_k^L\|^2 = \sum_{j=1}^{k-1} \tau_j^2 + (\eta_k \zeta_{k-1})^2 \quad \text{and} \quad \|x_k^C\|^2 = \sum_{j=1}^{k-1} \tau_j^2 + (\eta_k \zeta_{k-1} + \bar{\varepsilon}_k \bar{\zeta}_k)^2. \tag{28}$$

Both x_k^L and x_k^C may be found conveniently if we maintain the delayed iterate $\tilde{x}_{k-1} := \tau_1 v_1 + \cdots + \tau_{k-1} v_{k-1} = \tilde{x}_{k-2} + \tau_{k-1} v_{k-1}$, for then we have the orthogonal updates

$$x_k^L = \tilde{x}_{k-1} + \eta_k \zeta_{k-1} v_k \quad \text{and} \quad x_k^C = \tilde{x}_{k-1} + (\eta_k \zeta_{k-1} + \bar{\varepsilon}_k \bar{\zeta}_k) v_k. \tag{29}$$

PROPOSITION 2. *We have $\bar{\varepsilon}_1 \bar{\zeta}_1 = \tau_1$ and for $k > 1$, $\eta_k \zeta_{k-1} + \bar{\varepsilon}_k \bar{\zeta}_k = \tau_k$. This gives the same expressions as for standard CRAIG:*

$$x_k^C = \sum_{j=1}^{k} \tau_k v_k \quad \textit{and} \quad r_k^C = -\beta_{k+1} \tau_k u_{k+1}.$$

Proof. The identity for $k = 1$ follows from the definitions of $\bar{\varepsilon}_1$, $\bar{\zeta}_1$, and τ_1. By definition of $\bar{\zeta}_k$, we have $\bar{\varepsilon}_k \bar{\zeta}_k = \tau_k - \eta_k \zeta_{k-1}$, i.e., $\eta_k \zeta_{k-1} + \bar{\varepsilon}_k \bar{\zeta}_k = \tau_k$. The expressions for x_k^C and r_k^C follow from (29) and from (25), the definition of $\bar{\beta}_{k+1}$, and (18). ◻

Proposition 2 shows that x_k^C is updated along orthogonal directions, so that $\|x_k^C\|$ is monotonically increasing and $\|x_\star - x_k^C\|$ is monotonically decreasing, as stated by Paige (1974). Finally, (26) and Proposition 2 give $x_k^L = x_{k-1}^C + \eta_k \zeta_{k-1} v_k$.

Proposition 2 allows us to write $\tau_k - \eta_k \zeta_{k-1} = \bar{\epsilon}_k \bar{\zeta}_k$. Because $\beta_k \tau_{k-1} = -\alpha_k \tau_k$, the LQ residual may be rewritten

$$\begin{aligned} r_k^L &= \alpha_k (\tau_k - \eta_k \zeta_{k-1}) u_k - \bar{\beta}_{k+1} s_k \zeta_{k-1} u_{k+1} \\ &= \alpha_k \bar{\epsilon}_k \bar{\zeta}_k u_k - \alpha_k \beta_{k+1} s_k \zeta_{k-1} u_{k+1}, \end{aligned}$$

and correspondingly, $\|r_k^L\|^2 = \alpha_k^2((\bar{\epsilon}_k\bar{\zeta}_k)^2 + (\beta_{k+1}s_k\zeta_{k-1})^2)$. We are now able to establish a result that parallels Proposition 1.

PROPOSITION 3. *Let $x_\star$ and $y_\star$ be as in (1)–(3). The kth LNLQ iterates y_k^L and x_k^L solve*

$$\underset{y}{\text{minimize}}\ \|y - y_\star\| \ \text{ subject to }\ y \in \text{Range}(AA^T U_{k-1}), \tag{30}$$

$$\underset{x}{\text{minimize}}\ \|x - x_\star\|_{(AA^T)^\dagger} \ \text{ subject to }\ x \in \text{Range}(V_{k-1}), \tag{31}$$

respectively. In addition, y_k^L and x_k^L solve

$$\underset{y}{\text{minimize}}\ \|y\| \ \text{ subject to }\ y \in \text{Range}(U_k),\ b - AA^T y \perp \text{Range}(U_{k-1}), \tag{32}$$

$$\underset{x}{\text{minimize}}\ \|x\|_{(AA^T)^\dagger} \ \text{ subject to }\ x \in \text{Range}(V_k),\ b - Ax \perp \text{Range}(U_{k-1}). \tag{33}$$

Proof. By definition, $\bar{y}_k^L$ solves (16). Hence there must exist $\bar{t}$ such that $\bar{y}_k^L = H_{k-1}\bar{t}$ and $H_{k-1}^T\bar{y}_k^L = \beta_1 e_1$. By definition of H_{k-1} and (5), we have $y_k^L = U_k\bar{y}_k^L = U_k B_{k-1} L_{k-1}^T \bar{t} = AV_{k-1}L_{k-1}^T\bar{t} = AA^T U_{k-1}\bar{t}$.

The above implies that y_k^L is primal feasible for (30). Dual feasibility requires that $U_{k-1}^T AA^T(y - y_\star) = 0$, which is equivalent to $U_{k-1}^T r_k^L = 0$ because $AA^T y_\star = b$. The expression (24) confirms dual feasibility.

With $y_k^L \in \text{Range}(A)$, we have $y_k^L = (A^\dagger)^T x_k^L$ and then (31) follows from (30).

Using (24), we see that y_k^L is primal feasible for (32). Dual feasibility requires that $y_k^L = p + AA^T U_{k-1} q$ and $U_k^T p = 0$ for certain vectors p and q, but those conditions are satisfied for $p := 0$ and $q := \bar{t}$. Since $y_k^L = (A^\dagger)^T x_k^L$, we obtain (33) from (32). ∎

Note the subtle difference between the constraints of (14) and (33).

COROLLARY 1. *For each k, $\|x_k^C - x_\star\| \leqslant \|x_k^L - x_\star\|$.*

Proof. If we compare (12) with (31), we see that $\|x_k^C - x_\star\| \leqslant \|x_k^L - x_\star\|$ because $\text{Range}(V_{k-1}) \subset \text{Range}(V_k)$. ∎

3.5. Complete algorithm.

Algorithm 2 summarizes LNLQ. Note that if only the x part of the solution is desired, there is no need to initialize and update the vectors w_k, $\bar{w}_k$, y_k^L and y_k^C unless one wants to retrieve x as $A^T y$ at the end of the procedure. Similarly, if only the y part of the solution is desired, there is no need to initialize and update the vectors x_k^L and x_k^C. The update for x_{k+1}^C in line 18 of Algorithm 2 can be used even if the user wishes to dispense with updating x_k^L.

4. Regularization.

The regularized least-norm problem is

$$\underset{x\in\mathbb{R}^n,\, s\in\mathbb{R}^m}{\text{minimize}}\ \tfrac{1}{2}(\|x\|^2 + \|s\|^2) \quad \text{subject to }\ Ax + \lambda s = b, \tag{34}$$

which is compatible for any $\lambda \neq 0$. Saunders (1995, Result 7) states that applying Algorithm 1 to $\hat{A} := \begin{bmatrix} A & \lambda I \end{bmatrix}$ with initial vector b preserves U_k. We find corresponding $\widehat{V}_k$ and lower bidiagonal $\hat{L}_k$ by comparing the identities

$$\begin{bmatrix} A^T \\ \lambda I \end{bmatrix} U_k = \begin{bmatrix} V_k & \\ & U_k \end{bmatrix} \begin{bmatrix} L_k^T \\ \lambda I \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} A^T \\ \lambda I \end{bmatrix} U_k = \widehat{V}_k \hat{L}_k^T, \tag{35}$$

Algorithm 2 LNLQ

$\beta_1 u_1 = b, \;\; \alpha_1 v_1 = A^T u_1$ *begin Golub-Kahan process*
$\bar{\varepsilon}_1 = \alpha_1$, $\tau_1 = \beta_1/\alpha_1$, $\bar{\zeta}_1 = \tau_1/\bar{\varepsilon}_1$ *begin LQ factorization*
$w_1 = 0$, $\bar{w}_1 = u_1$
$y_1^L = 0$, $y_1^C = \bar{\zeta}_1 \bar{w}_1$
$x_1^L = 0$, $x_1^C = \tau_1 v_1$
for $k = 1, 2, \dots$ **do**
  $\beta_{k+1} u_{k+1} = A v_k - \alpha_k u_k$ *continue Golub-Kahan process*
  $\alpha_{k+1} v_{k+1} = A^T u_{k+1} - \beta_{k+1} v_k$
  $\varepsilon_k = (\bar{\varepsilon}_k^2 + \beta_{k+1}^2)^{\frac{1}{2}}$ *continue LQ factorization*
  $c_{k+1} = \bar{\varepsilon}_k/\varepsilon_k$, $s_{k+1} = \beta_{k+1}/\varepsilon_k$
  $\eta_{k+1} = \alpha_{k+1} s_{k+1}$, $\bar{\varepsilon}_{k+1} = -\alpha_{k+1} c_{k+1}$
  $\zeta_k = c_{k+1} \bar{\zeta}_k$, $\bar{\zeta}_{k+1} = (\tau_{k+1} - \eta_{k+1}\zeta_k)/\bar{\varepsilon}_{k+1}$ *prepare to update y*
  $w_k = c_{k+1} \bar{w}_k + s_{k+1} u_{k+1}$, $\bar{w}_{k+1} = s_{k+1} \bar{w}_k - c_{k+1} u_{k+1}$
  $y_{k+1}^L = y_k^L + \zeta_k w_k$ *update y*
  $y_{k+1}^C = y_{k+1}^L + \bar{\zeta}_{k+1} \bar{w}_{k+1}$
  $x_{k+1}^L = x_k^C + \eta_{k+1} \zeta_k v_{k+1}$ *update x*
  $\tau_{k+1} = -\beta_{k+1} \tau_k / \alpha_{k+1}$
  $x_{k+1}^C = x_k^C + \tau_{k+1} v_{k+1}$
end for

the first of which results from (5) and the second from Algorithm 1 applied to $\hat{A}$. At iteration k, we apply reflections $\hat{Q}_k$ designed to zero out the λI block, resulting in

$$\begin{bmatrix} V_k & \\ & U_k \end{bmatrix} \begin{bmatrix} L_k^T \\ \lambda I \end{bmatrix} = \begin{bmatrix} V_k & \\ & U_k \end{bmatrix} \hat{Q}_k^T \hat{Q}_k \begin{bmatrix} L_k^T \\ \lambda I \end{bmatrix} = \begin{bmatrix} \widehat{V}_k & \widehat{Y}_k \end{bmatrix} \begin{bmatrix} \hat{L}_k^T \\ 0 \end{bmatrix} = \widehat{V}_k \hat{L}_k^T.$$

Saunders (1995) uses $\hat{Q}_k$ to describe CRAIG with regularization under the name *extended CRAIG*. If we initialize $\lambda_1 := \lambda$, the first few reflections are illustrated in Figure 1, where shaded elements are those participating in the current reflection and grayed out elements have not yet been used. Two reflections per iteration are necessary, and the situation at iteration k may be described as

$$\begin{array}{c} \\ k \\ k+1 \end{array} \begin{array}{ccc} k & 2k & 2k+1 \\ \left[\begin{matrix} \alpha_k \\ \beta_{k+1} \end{matrix}\right. & \begin{matrix} \lambda_k \\ \; \end{matrix} & \left.\begin{matrix} \; \\ \lambda \end{matrix}\right] \end{array} \begin{array}{cc} k & 2k \\ \left[\begin{matrix} \hat{c}_k \\ \hat{s}_k \end{matrix}\right. & \left.\begin{matrix} \hat{s}_k \\ -\hat{c}_k \end{matrix}\right] \end{array} \begin{array}{cc} 2k & 2k+1 \\ \left[\begin{matrix} \tilde{c}_k \\ \tilde{s}_k \end{matrix}\right. & \left.\begin{matrix} \tilde{s}_k \\ -\tilde{c}_k \end{matrix}\right] \end{array} = \begin{array}{ccc} k & 2k & 2k+1 \\ \left[\begin{matrix} \hat{\alpha}_k \\ \hat{\beta}_{k+1} \end{matrix}\right. & \begin{matrix} 0 \\ \hat{\lambda}_{k+1} \end{matrix} & \left.\begin{matrix} \; \\ \lambda \end{matrix}\right] \end{array} \begin{array}{cc} 2k & 2k+1 \\ \left[\begin{matrix} \tilde{c}_k \\ \tilde{s}_k \end{matrix}\right. & \left.\begin{matrix} \tilde{s}_k \\ -\tilde{c}_k \end{matrix}\right] \end{array}$$

$$= \begin{array}{ccc} k & 2k & 2k+1 \\ \left[\begin{matrix} \hat{\alpha}_k \\ \hat{\beta}_{k+1} \end{matrix}\right. & \begin{matrix} 0 \\ 0 \end{matrix} & \left.\begin{matrix} \; \\ \lambda_{k+1} \end{matrix}\right] \end{array}.$$

The first reflection is defined by $\hat{\alpha}_k := \sqrt{\alpha_k^2 + \lambda_k^2}$, $\hat{c}_k := \alpha_k/\hat{\alpha}_k$, $\hat{s}_k := \lambda_k/\hat{\alpha}_k$, and results in $\hat{\beta}_{k+1} = \hat{c}_k \beta_{k+1}$ and $\hat{\lambda}_{k+1} = \hat{s}_k \beta_{k+1}$. The second reflection defines $\lambda_{k+1} := \sqrt{\hat{\lambda}_{k+1}^2 + \lambda^2}$, $\tilde{c}_k := \hat{\lambda}_{k+1}/\lambda_{k+1}$, $\tilde{s}_k := \lambda/\lambda_{k+1}$, and does not create a new nonzero.

$$
\begin{bmatrix}
\alpha_1 & & & & \lambda_1 & & & \\
\beta_2 & \alpha_2 & & & & \lambda & & \\
 & \beta_3 & \alpha_3 & & & & \lambda & \\
 & & \beta_4 & \alpha_4 & & & & \lambda
\end{bmatrix}
\to
\begin{bmatrix}
\hat\alpha_1 & & & & 0 & & & \\
\hat\beta_2 & \alpha_2 & & & \hat\lambda_2 & \lambda & & \\
 & \beta_3 & \alpha_3 & & & & \lambda & \\
 & & \beta_4 & \alpha_4 & & & & \lambda
\end{bmatrix}
$$

$$
\to
\begin{bmatrix}
\hat\alpha_1 & & & & 0 & & & \\
\hat\beta_2 & \alpha_2 & & & 0 & \lambda_2 & & \\
 & \beta_3 & \alpha_3 & & & & \lambda & \\
 & & \beta_4 & \alpha_4 & & & & \lambda
\end{bmatrix}
\to
\begin{bmatrix}
\hat\alpha_1 & & & & 0 & & & \\
\hat\beta_2 & \hat\alpha_2 & & & 0 & 0 & & \\
 & \hat\beta_3 & \alpha_3 & & & \hat\lambda_3 & \lambda & \\
 & & \beta_4 & \alpha_4 & & & & \lambda
\end{bmatrix}
$$

$$
\to
\begin{bmatrix}
\hat\alpha_1 & & & & 0 & & & \\
\hat\beta_2 & \hat\alpha_2 & & & 0 & 0 & & \\
 & \hat\beta_3 & \alpha_3 & & & 0 & \lambda_3 & \\
 & & \beta_4 & \alpha_4 & & & & \lambda
\end{bmatrix}
\to
\begin{bmatrix}
\hat\alpha_1 & & & & 0 & & & \\
\hat\beta_2 & \hat\alpha_2 & & & 0 & 0 & & \\
 & \hat\beta_3 & \hat\alpha_3 & & & 0 & 0 & \\
 & & \hat\beta_4 & \alpha_4 & & & \hat\lambda_4 & \lambda
\end{bmatrix}
$$

Fig. 1. *Illustration of a few steps of the factorization in the presence of regularization.*

Only the first reflection contributes to the kth column of $\widehat{V}_k$:

$$
\begin{array}{c} \begin{array}{cc} {\scriptstyle k} & {\scriptstyle 2k} \end{array} \\ \begin{bmatrix} v_k & 0 \\ 0 & u_k \end{bmatrix} \end{array}
\begin{array}{c} \begin{array}{cc} {\scriptstyle k} & {\scriptstyle 2k} \end{array} \\ \begin{bmatrix} \hat c_k & \hat s_k \\ \hat s_k & -\hat c_k \end{bmatrix} \end{array}
=
\begin{array}{c} \begin{array}{cc} {\scriptstyle k} & {\scriptstyle 2k} \end{array} \\ \begin{bmatrix} \hat c_k v_k & \hat s_k v_k \\ \hat s_k u_k & -\hat c_k u_k \end{bmatrix} \end{array}. \tag{36}
$$

Iteration k of LNLQ with regularization solves (16), but H_{k-1}^T is then the top $(k-1)\times k$ submatrix of

$$
\begin{bmatrix} L_k & \lambda I \end{bmatrix} \begin{bmatrix} L_k^T \\ \lambda I \end{bmatrix} = L_k L_k^T + \lambda^2 I = T_k + \lambda^2 I.
$$

In (17), we compute the LQ factorization of $\hat L_k^T$ instead of L_k^T, but the details are identical, as are the updates of y_k^L in (21) and y_k^C in (22). Because U_k is unchanged by regularization, the residual expressions (24) and (25) remain valid. Subsequently,

$$
\begin{bmatrix} x_k^L \\ s_k^L \end{bmatrix} = \begin{bmatrix} A^T \\ \lambda I \end{bmatrix} U_k \bar y_k = \widehat{V}_k \hat L_k^T \bar y_k,
$$

but we are only interested in the top half, x_k^L. Let the top $n \times k$ submatrix of $\widehat{V}_k$ be

$$
\widehat{W}_k := \begin{bmatrix} \widehat{w}_1 & \cdots & \widehat{w}_k \end{bmatrix} = \begin{bmatrix} I & 0 \end{bmatrix} \widehat{V}_k = \begin{bmatrix} V_k & 0 \end{bmatrix} \hat Q_k^T.
$$

We conclude from (36) that $\widehat{w}_j = \hat c_j v_j$ for $j = 1, \dots, k$. The update (27) remains valid with v_k replaced by $\widehat{w}_k$.

5. Error upper bounds.

5.1. Upper bound on $\|y_\star - y_k^L\|$. By orthogonality, $\|y_\star - y_k^L\|^2 = \|y_\star\|^2 - \|y_k^L\|^2$. If A has full row rank, $y_\star = (AA^T)^{-1} b$ and $\|y_\star\|^2 = b^T (AA^T)^{-2} b$. If we define

$$
f(AA^T) := \sum_{i=1}^m f(\sigma_i^2) q_i q_i^T
$$

for any given $f : (0, \infty) \to \mathbb{R}$, where q_i is the ith left singular vector of A, then $\|y_\star\|^2 = b^T f(AA^T) b$ with $f(\xi) := \xi^{-2}$. More generally, as $y_\star$ is the minimum-norm solution of (3), it may be expressed as

$$y_\star = \sum_{i=r}^{m} f(\sigma_i^2)\,(q_i^T b)\,q_i,$$

where σ_r is the smallest nonzero singular value of A, which amounts to redefining $f(\xi) := 0$ at $\xi = 0$. Because $b = \beta_1 u_1$, we may write

$$\|y_\star\|^2 = \beta_1^2 \sum_{i=1}^{m} f(\sigma_i^2)\mu_i^2, \qquad \mu_i := q_i^T u_1,\ i = 1, \dots, m.$$

We obtain an upper bound on $\|y_\star\|$ by viewing the sum as a Riemann-Stieltjes integral for a well chosen Stieltjes measure and approximating the integral via a Gauss-Radau quadrature. We refer to Golub and Meurant (2010) for background.

The fixed Gauss-Radau quadrature node is set to a prescribed $\sigma_{\text{est}} \in (0, \sigma_r)$. We follow Estrin et al. (2019b) and modify L_k rather than T_k. Let

$$\widetilde{L}_k := \begin{bmatrix} L_{k-1} & 0 \\ \beta_k e_{k-1}^T & \omega_k \end{bmatrix}, \tag{37}$$

which differs from L_k in its (k, k)th element only, and

$$\widetilde{T}_k := \widetilde{L}_k \widetilde{L}_k^T = \begin{bmatrix} T_{k-1} & \bar{\beta}_{k-1} e_{k-1} \\ \bar{\beta}_{k-1} e_{k-1}^T & \beta_k^2 + \omega_k^2 \end{bmatrix}$$

(with $\bar{\beta}_{k-1}$ defined in (10)), which differs from T_k in its (k, k)th element only. The Poincaré separation theorem ensures that the singular values of L_k lie in (σ_r, σ_1). The Cauchy interlace theorem for singular values ensures that it is possible to select ω_k so that the smallest singular value of (37) is σ_{est}.

The next result derives from (Golub and Meurant, 1994, Theorems 3.2 and 3.4).

Theorem 1 (Estrin et al., 2019b, Theorem 4). *Let $f : [0, \infty) \to \mathbb{R}$ be such that $f^{(2j+1)}(\xi) < 0$ for all $\xi \in (\sigma_r^2, \sigma_1^2)$ and all $j \geqslant 0$. Fix $\sigma_{est} \in (0, \sigma_r)$. Let L_k be the bidiagonal generated after k steps of Algorithm 1, and $\omega_k > 0$ be chosen so that the smallest singular value of (37) is σ_{est}. Then,*

$$b^T f(AA^T) b \leqslant \beta_1^2 e_1^T f(\widetilde{L}_k \widetilde{L}_k^T) e_1.$$

The procedure for identifying ω_k is identical to that of Estrin et al. (2019b) and yields $\omega_k = \sqrt{\sigma_{\text{est}}^2 - \sigma_{\text{est}} \beta_k \theta_{2k-2}}$, where θ_{2k-2} is an element of a related eigenvector.

Application of Theorem 1 to $f(\xi) := \xi^{-2}$ with the convention that $f(0) := 0$ provides an upper bound on $\|y_\star\|^2$.

Corollary 2. *Fix $\sigma_{est} \in (0, \sigma_r)$. Let L_k be the bidiagonal generated after k steps of Algorithm 1, and $\omega_k > 0$ be chosen so that the smallest singular value of (37) is σ_{est}. Then*

$$\|y_\star\|^2 \leqslant \beta_1^2 e_1^T \big(\widetilde{L}_k \widetilde{L}_k^T\big)^{-2} e_1.$$

To evaluate the bound in Corollary 2, we modify the LQ factorization (17) to

$$\widetilde{L}_k^T = \begin{bmatrix} L_{k-1}^T & \beta_k e_{k-1} \\ 0 & \omega_k \end{bmatrix} = \begin{bmatrix} M_{k-1} & \\ \widetilde{\eta}_k e_{k-1}^T & \widetilde{\varepsilon}_k \end{bmatrix} \begin{bmatrix} Q_{k-1} & \\ & 1 \end{bmatrix} = \widetilde{M}_k Q_k,$$

where $\widetilde{\eta}_k = \omega_k s_k$ and $\widetilde{\varepsilon}_k = -\omega_k c_k$. Define $\widetilde{t}_k$ and $\widetilde{z}_k$ from

$$\text{(38)} \qquad \widetilde{L}_k \widetilde{t}_k = \beta_1 e_1 \quad \text{and} \quad \widetilde{M}_k \widetilde{z}_k = \widetilde{t}_k.$$

The updated factorization and the definition of f yield

$$\|y_\star\|^2 \leqslant \beta_1^2 \|(\widetilde{L}_k \widetilde{M}_k Q_k)^{-1} e_1\|^2 = \beta_1^2 \|\widetilde{M}_k^{-1} \widetilde{L}_k^{-1} e_1\|^2 = \|\widetilde{M}_k^{-1} \widetilde{t}_k\|^2 = \|\widetilde{z}_k\|^2.$$

Comparing with the definition of t_k and z_k in subsection 3.2 reveals that $\widetilde{t}_k = (t_{k-1}, \widetilde{\tau}_k)$ and $\widetilde{z}_k = (z_{k-1}, \widetilde{\zeta}_k)$, with $\widetilde{\tau}_k = -\beta_k \tau_{k-1}/\omega_k$ and $\widetilde{\zeta}_k = (\widetilde{\tau}_k - \widetilde{\eta}_k \zeta_{k-1})/\widetilde{\varepsilon}_k$. Combining with (23) yields the bound

$$\text{(39)} \qquad \|y_\star - y_k^L\|^2 = \|y_\star\|^2 - \|z_{k-1}\|^2 \leqslant \|z_{k-1}\|^2 + \widetilde{\zeta}_k^2 - \|z_{k-1}\|^2 = \widetilde{\zeta}_k^2.$$

5.2. Upper bound on $\|y_\star - y_k^C\|$. Estrin et al. (2019a, Theorem 6) establishes that $\|y_\star - y_k^C\| \leqslant \|y_\star - y_k^L\|$, so that the bound from the previous section applies. With $\bar{\zeta}_k$ defined in subsection 3.2, Estrin et al. (2019a) derive the improved bound

$$\text{(40)} \qquad \|y_\star - y_k^C\|^2 \leqslant \widetilde{\zeta}_k^2 - \bar{\zeta}_k^2.$$

They provide further refinement of this bound by using the sliding window approach of Golub and Strakŏs (1994). For a chosen delay d, $O(d)$ scalars can be stored at each iteration, and for $O(d)$ additional work, a quantity $\theta_k^{(d)} \geqslant 0$ can be computed so that

$$\text{(41)} \qquad \|y_\star - y_k^C\|^2 \leqslant \widetilde{\zeta}_k^2 - \bar{\zeta}_k^2 - 2\theta_k^{(d)}.$$

The definitions of c_k, s_k, ζ_k, and $\bar{\zeta}_k$ match those of Estrin et al. (2019a).

5.3. Upper bound on $\|x_\star - x_k^C\|$. Assume temporarily that A has full row rank. By orthogonality in (26), $\|x_\star - x_k^C\|^2 = \|x_\star\|^2 - \|x_k^C\|^2$. We may then use

$$\|x_\star\|^2 = \|A^T y_\star\|^2 = \|y_\star\|^2_{AA^T} = \|b\|^2_{(AA^T)^{-1}}.$$

Applying Theorem 1 to $f(\xi) := \xi^{-1}$ with $f(0) := 0$ provides an upper bound on $\|x_\star\|^2$ in the vein of Golub and Meurant (1994, Theorems 3.2 and 3.4).

Corollary 3. *Fix $\sigma_{est} \in (0, \sigma_r)$. Let L_k be the bidiagonal generated after k steps of Algorithm 1, and $\omega_k > 0$ be chosen so that the smallest singular value of (37) is σ_{est}. Then*

$$\|x_\star\|^2 \leqslant \beta_1^2 e_1^T (\widetilde{L}_k \widetilde{L}_k^T)^{-1} e_1.$$

We use (38) to evaluate the bound of Corollary 3 as

$$\beta_1^2 e_1^T (\widetilde{L}_k \widetilde{L}_k^T)^{-1} e_1 = \|\beta_1 \widetilde{L}_k^{-1} e_1\|^2 = \|\widetilde{t}_k\|^2,$$

which leads to the bound

$$\text{(42)} \qquad \|x_\star - x_k^C\|^2 \leqslant \|\widetilde{t}_k\|^2 - \|t_k\|^2 = \widetilde{\tau}_k^2 - \tau_k^2.$$

This coincides with the bound of Arioli (2013), who derived it using the Cholesky factorization of T_k.

Note that Arioli (2013, Equation (4.4)) proposes the error bound

$$\text{(43)} \qquad \|y_\star - y_k^C\| = \|L_n^{-1}(x_\star - x_k^C)\| \leqslant \sigma_{\min}(L_k)^{-1} \|x_\star - x_k^C\| \leqslant \sigma_r^{-1} \|x_\star - x_k^C\|.$$

It may be possible to improve on (43) by maintaining a running estimate of $\sigma_{\min}(L_k)$, such as the estimate $\min(\varepsilon_1, \dots, \varepsilon_{k-1}, \bar{\varepsilon}_k)$ discussed by Stewart (1999).

5.4. Upper bound on $\|\boldsymbol{x}_\star - \boldsymbol{x}_k^L\|$. Using $x_k^L = x_{k-1}^C + \eta_k \zeta_{k-1} v_k$, we have

$$\|x_\star - x_k^L\|^2 = \left\| V_n \left(t_n - \begin{bmatrix} t_{k-1} \\ \eta_k \zeta_{k-1} \\ 0 \end{bmatrix} \right) \right\|^2 = \|x_\star - x_k^C\|^2 + (\tau_k - \eta_k \zeta_{k-1})^2.$$

Thus, using the error bound in (42) we obtain

$$\|x_\star - x_k^L\|^2 \leqslant \tilde{\tau}_k^2 - \tau_k^2 + (\tau_k - \eta_k \zeta_{k-1})^2. \tag{44}$$

5.5. Choice of σ_{est}. We briefly discuss choosing σ_{est} and its effect on the error upper bounds. When A is symmetric positive definite, numerical experiments in Estrin et al. (2019a, §8.4) show the effect of σ_{est} on the error bound quality; similar trends are observed for LNLQ and CRAIG, so we do not repeat such experiments here.

Estrin et al. (2019a, §6) also discuss aspects of obtaining an eigenvalue estimate (in this case, a singular value estimate). Being able to obtain σ_{est} is often application-dependent and good estimates may not be available in general; in such cases, many Gauss-Radau-based estimation procedures (such as the one here) may not be applicable. In some cases, σ_{est} is readily available, e.g., if the problem is regularized, or via a preconditioning approach (see subsection 7.2).

Meurant and Tichý (2018) provide a Gauss-Radau-based error estimation procedure for CG that at every iteration uses a cheap estimate of the smallest Ritz value as the eigenvalue estimate. The advantage is that lower bounds on the spectrum of A do not need to be known a priori, but because the smallest Ritz value is not a lower bound, the resulting estimates are not guaranteed to be upper bounds. However, the resulting bounds are shown to be effective in practice. A future avenue of work is to adapt this approach to our error estimation procedure to avoid requiring a readily available singular value underestimate.

6. Preconditioning. As with other Golub-Kahan-based methods, convergence depends on the distribution of $\{\sigma_i(A)\}$. Therefore we consider an equivalent system $N^{-\frac{1}{2}} A A^T N^{-\frac{1}{2}} N^{\frac{1}{2}} y = N^{-\frac{1}{2}} b$, where $N^{-\frac{1}{2}} A$ has clustered singular values.

For the unregularized problem (3), to run preconditioned LNLQ efficiently we replace Algorithm 1 by the generalized Golub-Kahan process (Arioli, 2013, Algorithm 3.1). We seek a preconditioner $N \succ 0$ such that $N \approx AA^T$, and require no changes to the algorithm except in how we generate vectors u_k and v_k. This is equivalent to applying a block-diagonal preconditioner to the saddle-point system

$$\begin{bmatrix} I & \\ & N^{-1} \end{bmatrix} \begin{bmatrix} -I & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} I & \\ & N^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ b \end{bmatrix}.$$

For a regularized system with $\lambda \neq 0$, we need to solve a 2×2 quasi-definite system

$$\begin{bmatrix} -I & A^T \\ A & \lambda^2 I \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix}. \tag{45}$$

We cannot directly precondition with generalized Golub-Kahan as before, because properties analogous to (35) do not hold for $N \neq I$. Instead we must precondition the equivalent 3×3 block system

$$\begin{bmatrix} I & & \\ & I & \\ & & N^{-1} \end{bmatrix} \begin{bmatrix} -I & & A^T \\ & -I & \lambda I \\ A & \lambda I & \end{bmatrix} \begin{bmatrix} x \\ s \\ y \end{bmatrix} = \begin{bmatrix} I & & \\ & I & \\ & & N^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ b \end{bmatrix},$$

where $N \approx AA^T + \lambda^2 I$ is a symmetric positive definite preconditioner. In effect, we must run preconditioned LNLQ directly on $\hat{A} = \begin{bmatrix} A & \lambda I \end{bmatrix}$.

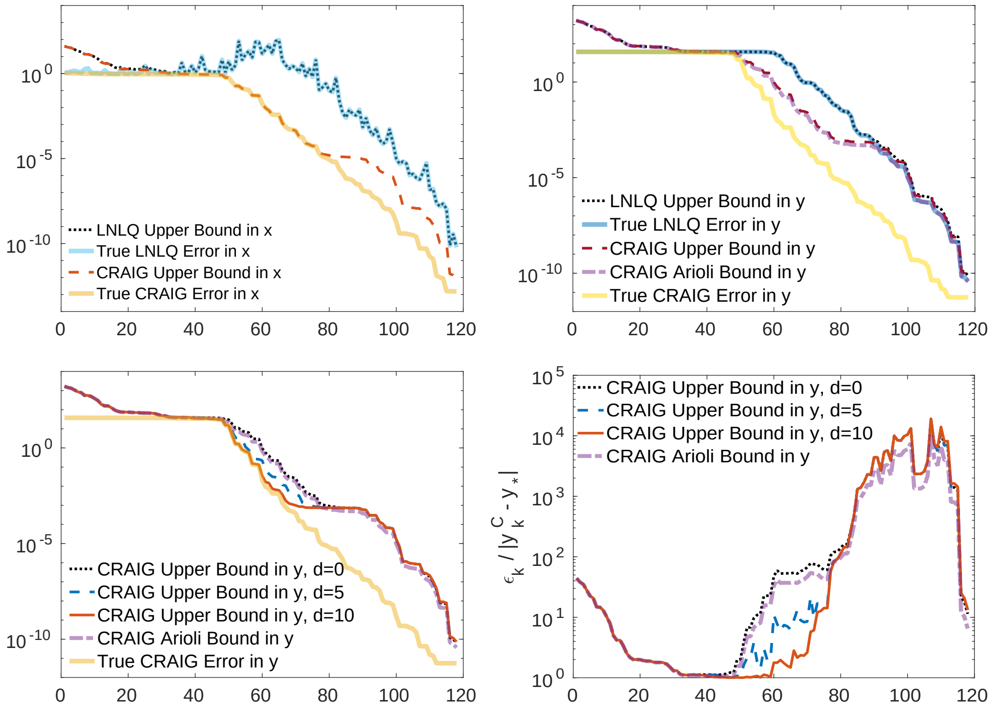

FIG. 2. *Error in x_k (top left) and y_k (top right) along the LNLQ and CRAIG iterations for Meszaros/scagr7-2c. The solid blue (yellow) line is the exact error for LNLQ (CRAIG), and the remaining lines show the various error bounds. The bottom left plot shows the improved bounds* (41) *and bounds from* Arioli *(*2013*) for the error in y_k for CRAIG with $d = 5$ and 10. The bottom right plot shows the same bounds divided by the true error.*

7. Implementation and numerical experiments.

We implemented LNLQ in Matlab[1], including the relevant error bounds. The exact solution for each experiment is computed using Matlab's backslash operator on the augmented system (3). Mentions of CRAIG below refer to transferring from the LNLQ point to the CRAIG point.

7.1. UFL problems.

Matrix Meszaros/scagr7-2c from the UFL collection (Davis and Hu, 2011) has size 2447 × 3479. We set $b = e/\sqrt{m}$, the normalized vector of ones. For LNLQ and CRAIG we record the error in x_k and y_k at each iteration using the exact solution, and the error bounds discussed above using $\sigma_{\text{est}} = (1 - 10^{-10})\,\sigma_{\min}(A)$, where $\sigma_{\min}(A)$ was provided by the UFL collection. The same σ_{est} is used to evaluate the bound (43). Figure 2 records the results.

We see that the LNLQ error bounds are tight, even though the error in x_k is not monotonic. In accordance with Proposition 1, the CRAIG error in x_k is lower than the LNLQ error. The same for the error in y_k. The CRAIG error in x_k is tight until the Gauss-Radau quadrature becomes inaccurate—a phenomenon also observed by Meurant and Tichý (2014); Meurant and Tichý (2018).

Regarding the CRAIG error in y_k, we see that the error bounds from (40) and (43) are close to each other, with (43) being slightly tighter. We observed that the simpler bound (43) nearly overlaps with the bound (40) on other problems. However, (41) provides the ability to tighten (40), and even small delays such as $d = 5$ or 10 can

[1] Available from github.com/restrin/LinearSystemSolvers

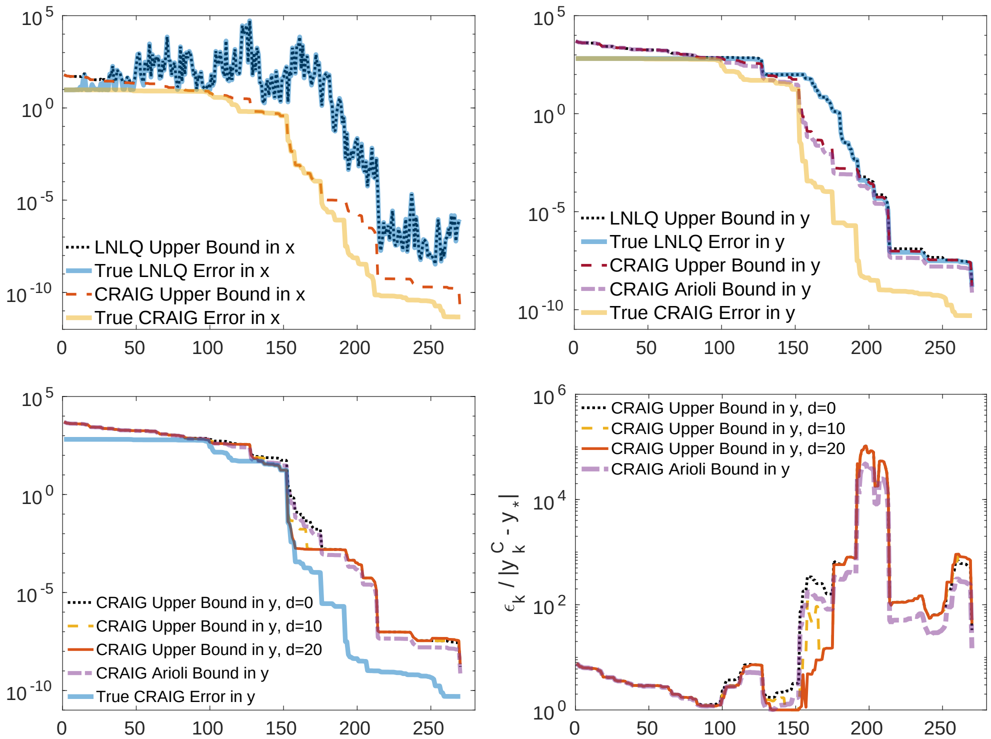

Fig. 3. *Error in x_k (top left) and y_k (top right) along the LNLQ and CRAIG iterations for LPnetlib/lp_kb2. The solid blue (yellow) line is the exact error for LNLQ (CRAIG), and the remaining lines show the various error bounds. The bottom left plot shows the improved bounds* (41) *and bounds from Arioli* (2013) *for the error in y_k for CRAIG with $d = 5$ and 10. The bottom right plot shows the same bounds divided by the true error.*

improve the bound significantly until the Gauss-Radau quadrature becomes inaccurate. Thus, the sliding window approach can be useful when an accurate estimate of $\sigma_{\min}(A)$ is available and early termination is relevant, for example when only a crude approximation of $x_\star$ and $y_\star$ is required.

In Figure 3 we repeat the experiment with UFL problem LPnetlib/lp_kb2, which has size 43×68. Because LNLQ and CRAIG take more than 250 iterations, it is clear that global orthogonality is violated, yet the upper bounds remain faithful. Hence, it may be possible to derive these bounds by assuming only local orthogonality in the Golub-Kahan process. This is a direction for future research.

7.2. Fletcher's penalty function. We now apply LNLQ to least-norm problems arising from using Fletcher's exact penalty function (Fletcher, 1973; Estrin et al., 2018) to solve PDE-constrained control problems. We consider the problem

$$\begin{aligned} &\underset{\mathbf{u},\,\mathbf{z}}{\text{minimize}} && \tfrac{1}{2}\int_\Omega \|\mathbf{u}-\mathbf{u}_d\|^2\,\mathrm{d}x + \tfrac{1}{2}\alpha\int_\Omega \mathbf{z}^2\,\mathrm{d}x \\ &\text{subject to} && \nabla\cdot(\mathbf{z}\nabla\mathbf{u}) = -\sin(\omega x_1)\sin(\omega x_2) \quad \text{in } \Omega, \\ & && \mathbf{u} = 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{46}$$

where $\omega = \pi - \frac{1}{8}$, $\Omega = [-1,1]^2$, and $\alpha = 10^{-4}$ is a small regularization parameter. Here, $\mathbf{u}$ might represent the temperature distribution on a square metal plate, $\mathbf{u}_d$ is

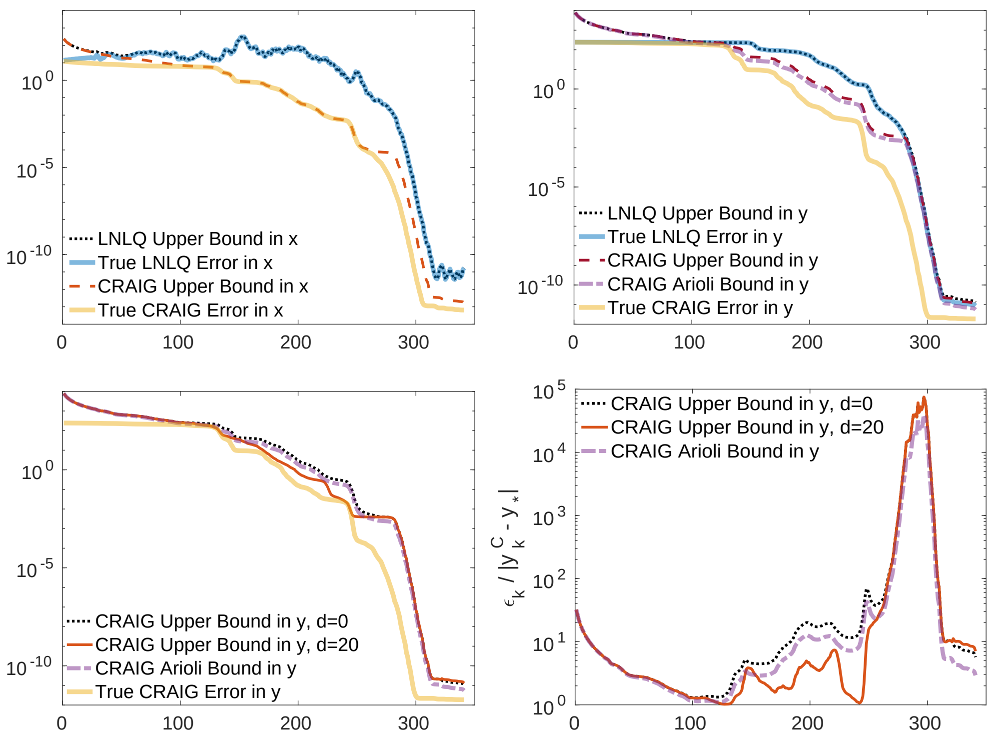

Fig. 4. *Error in x_k (top left) and y_k (top right) along the LNLQ and CRAIG iterations. The solid blue (yellow) line is the exact error for LNLQ (CRAIG), and the remaining lines show the various error bounds. The bottom left plot shows the improved bounds* (41) *and bounds from Arioli (2013) for the error in y_k for CRAIG with $d = 20$. The bottom right plot shows the same bounds divided by the true error.*

the observed temperature, and we aim to determine the diffusion coefficients $\mathbf{z}$ so that $\mathbf{u}$ matches the observations in a least-squares sense. We discretize (46) using finite elements with triangular cells, and obtain the equality-constrained problem

$$\underset{\bar{u}}{\text{minimize}}\ \ f(\bar{u}) \ \text{ subject to } \ c(\bar{u}) = 0.$$

Let p be the number of cells along one dimension, so that $u \in \mathbb{R}^{p^2}$ and $z \in \mathbb{R}^{(p+2)^2}$ are the discretizations of $\mathbf{u}$ and $\mathbf{z}$, $\bar{u} := (u, z)$, and $c(\bar{u}) \in \mathbb{R}^{p^2}$. We use $p = 31$ in the experiments below. Let $A(\bar{u}) := \begin{bmatrix} A_u & A_z \end{bmatrix}$ be the Jacobian of $c(\bar{u})$.

For a given penalty parameter $\sigma > 0$, Fletcher's exact penalty approach is to

$$\begin{aligned} &\underset{\bar{u}}{\text{minimize}}\ \ \phi_\sigma(\bar{u}) := f(\bar{u}) - c(\bar{u})^T y_\sigma(\bar{u}) \\ &\qquad \text{where} \ \ y_\sigma(\bar{u}) \in \arg\min_y \tfrac{1}{2} \left\| \nabla f(\bar{u}) - A(\bar{u})^T y \right\|^2 + \sigma c(\bar{u})^T y. \end{aligned}$$

In order to evaluate $\phi_\sigma(\bar{u})$ and $\nabla \phi_\sigma(\bar{u})$, we must solve systems of the form (3). For these experiments, we use $b = -c(\bar{u})$ and $A = A(\bar{u})$. Note that by controlling the error in the solution of (3), we control the inexactness in the computation of the penalty function value and gradient. In our experiments, we evaluate b and A at $\bar{u} = e$, the vector of ones. We first apply LNLQ and CRAIG without preconditioning. The results are summarized in Figure 4.

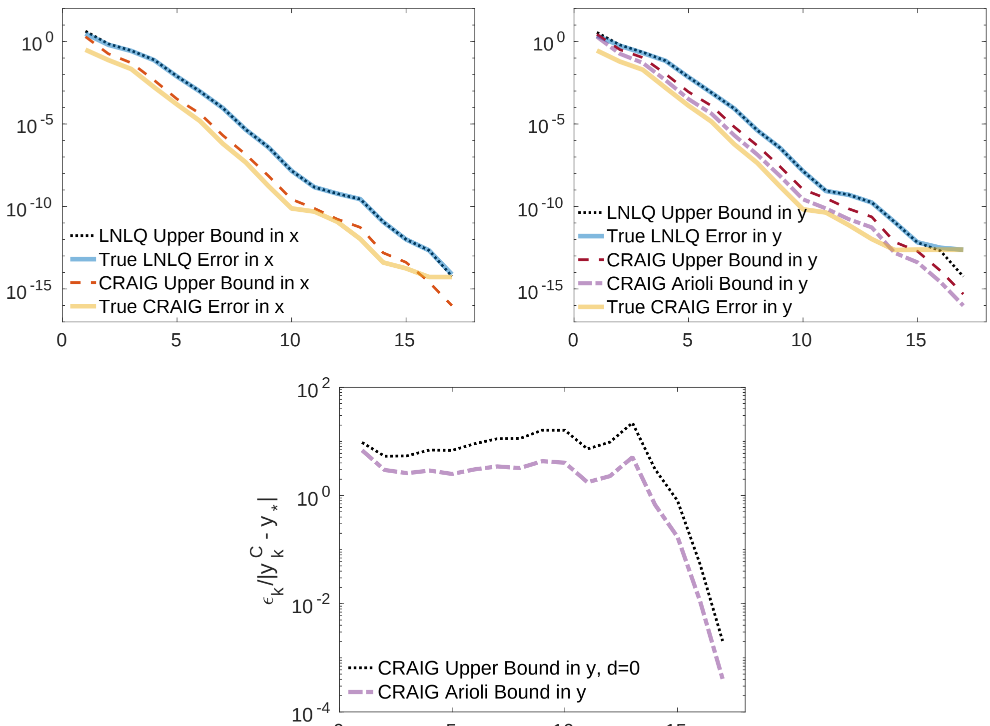

FIG. 5. *Error in x_k (left) and y_k (right) along the LNLQ and CRAIG iterations. The solid blue (yellow) line is the exact error for LNLQ (CRAIG), and the remaining lines show the various error bounds. The bottom plot shows the same bounds for CRAIG for the error in y_k, but divided by the true error.*

We observe trends like those in the previous section. The LNLQ bounds are quite accurate because of our good estimate of the smallest singular value, even though the LNLQ error in x_k is not monotonic. The CRAIG error bound for x_k is tight until the Gauss-Radau quadrature becomes inaccurate, which results in a looser bound. The latter impacts the CRAIG error bound for y_k in the form of the plateau after iteration 250. The error bound (43) is slightly tighter than (40), while if we use (41) with $d = 20$, we achieve a tighter bound until the plateau occurs.

We now use the preconditioner $N = A_u A_u^T$, which corresponds to two solves of Poisson's equation with fixed diffusion coefficients. Because $\sigma_{\min}((A_u A_u)^{-1} A A^T) = \sigma_{\min}(I + (A_u A_u^T)^{-1} A_z A_z^T) \geqslant 1$, we choose $\sigma_{\text{est}} = 1$. Recall that the y-error is now measured in the N-energy norm. The results appear in Figure 5.

We see that the preconditioner is effective, and that $\sigma_{\text{est}} = 1$ is an accurate approximation as the LNLQ error bounds are extremely tight. The CRAIG error bounds are tight as well, although the error "bounds" for y_k go below the true error in the last few iterations, which is expected and observed in Estrin et al. (2019a).

8. Extension to symmetric quasi-definite systems. Given symmetric and positive definite M and N whose inverses can be applied efficiently, LNLQ generalizes to the solution of the symmetric quasi-definite system (Vanderbei, 1995)

$$\mathcal{K}\begin{bmatrix} x \\ y \end{bmatrix} := \begin{bmatrix} M & A^T \\ A & -N \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix}, \tag{47}$$

which represents the optimality conditions of both problems

$$\underset{x,\,y}{\text{minimize}}\ \tfrac{1}{2}\|x\|_M^2 + \tfrac{1}{2}\|y\|_N^2 \quad \text{subject to}\ \ Ax - Ny = b, \tag{48}$$

$$\underset{x}{\text{minimize}}\ \tfrac{1}{2}\|Ax - b\|_{N^{-1}}^2 + \tfrac{1}{2}\|x\|_M^2. \tag{49}$$

The only changes required are to substitute Algorithm 1 for the generalized Golub-Kahan process (Orban and Arioli, 2017, Algorithm 4.2) and to set the regularization parameter $\lambda := 1$. This requires one system solve with M and one system solve with N per iteration.

Applying LSLQ (Estrin et al., 2019b) to (49) is implicitly equivalent to applying SYMMLQ to the normal equations

$$(A^TN^{-1}A + M)x = A^TN^{-1}b, \tag{50}$$

while applying LNLQ to (48) is equivalent to applying SYMMLQ to the normal equations of the second kind:

$$(AM^{-1}A^T + N)y = c, \qquad Mx = A^Ty, \tag{51}$$

where we changed the sign of y to avoid distracting minus signs.

In lieu of (5), the generalized Golub-Kahan process can be summarized as

$$AV_k = MU_{k+1}B_k, \tag{52a}$$

$$A^TU_{k+1} = NV_kB_k^T + \alpha_{k+1}Nv_{k+1}e_{k+1}^T = NV_{k+1}L_{k+1}^T, \tag{52b}$$

where now $U_k^TMU_k = I$ and $V_k^TNV_k = I$ in exact arithmetic. Pasting (52) together yields

$$\begin{bmatrix} M & A^T \\ A & -N \end{bmatrix}\begin{bmatrix} V_k & \\ & U_k \end{bmatrix} = \begin{bmatrix} M & \\ & N \end{bmatrix}\begin{bmatrix} V_k & \\ & U_k \end{bmatrix}\begin{bmatrix} I & L_k^T \\ L_k & -I \end{bmatrix} + \begin{bmatrix} 0 \\ \beta_{k+1}Nu_{k+1} \end{bmatrix} e_{2k}^T,$$

$$\begin{bmatrix} M & A^T \\ A & -N \end{bmatrix}\begin{bmatrix} V_k & \\ & U_{k+1} \end{bmatrix} = \begin{bmatrix} M & \\ & N \end{bmatrix}\begin{bmatrix} V_k & \\ & U_{k+1} \end{bmatrix}\begin{bmatrix} I & B_k^T \\ B_k & -I \end{bmatrix} + \begin{bmatrix} \alpha_{k+1}Mv_{k+1} \\ 0 \end{bmatrix} e_{2k+1}^T.$$

These relations correspond to a Lanczos process applied to (47) with preconditioner blkdiag$(M,\,N)$. The small symmetric quasi-definite matrix on the right-hand side of the preceding identities is a symmetric permutation of the Lanczos tridiagonal, which is found by restoring the order in which the Lanczos vectors $(v_k,\,0)$ and $(0,\,u_k)$ are generated:

$$T_{2k+1} = \begin{bmatrix} 1 & \alpha_1 & & & & \\ \alpha_1 & -1 & \beta_2 & & & \\ & \beta_2 & 1 & \ddots & & \\ & & \ddots & \ddots & \alpha_k & \\ & & & \alpha_k & -1 & \beta_{k+1} \\ & & & & \beta_{k+1} & 1 \end{bmatrix} = \begin{bmatrix} T_{2k} & \beta_{k+1}e_{2k} \\ \beta_{k+1}e_{2k}^T & 1 \end{bmatrix}.$$

Saunders (1995) and Orban and Arioli (2017) show that the CG iterates are well-defined for (47) even though $\mathcal{K}$ is indefinite. In a similar vein, Orban and Arioli (2017) establish that applying MINRES to (47) with the block-diagonal preconditioner produces alternating preconditioned LSMR and LSQR iterations, where LSMR is applied to (50) and LSQR is applied to (51).

It turns out that SYMMLQ applied directly to (47) with this preconditioner satisfies the following property: even iterations are CG iterations, while odd iterations take a zero step and make no progress. Thus every other iteration is wasted. The generalized iterative methods of Orban and Arioli (2017), LSLQ or LNLQ should be used instead. The property is formalized in the following result.

THEOREM 2. *Let x_k^{LQ} and x_k^{CG} be the iterates generated at iteration k of SYMMLQ and CG applied to (47), and x_k^C be the iterate defined in (7). Then for $k \geqslant 1$, $x_{2k-1}^{LQ} = x_{2k}^{LQ} = x_{2k}^{CG} = x_k^C$.*

Proof. For brevity, we use the notation from (Estrin et al., 2019a, §2.1) to describe the Lanczos process and how to construct the CG and SYMMLQ iterates. By (52), $\underline{T}_k$ and the L factor of the LQ factorization of $\underline{T}_{k-1}^T$ have the form

$$\underline{T}_k = \begin{bmatrix} 1 & t_2 & & & \\ t_2 & -1 & t_3 & & \\ & t_3 & 1 & \ddots & \\ & & \ddots & \ddots & t_k \\ & & & t_k & (-1)^{k-1} \\ & & & & t_{k+1} \end{bmatrix}, \qquad L_k = \begin{bmatrix} \gamma_1 & & & & \\ \delta_2 & \gamma_2 & & & \\ \varepsilon_3 & \delta_3 & \gamma_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \varepsilon_{k-1} & \delta_{k-1} & \gamma_{k-1} \end{bmatrix},$$

where each t_i is a scalar. For $k \geqslant 2$, the LQ factorization is accomplished using reflections defined by

$$\begin{bmatrix} \bar{\gamma}_{k-1} & t_k \\ \bar{\delta}_k & (-1)^{k-1} \\ 0 & t_{k+1} \end{bmatrix} \begin{bmatrix} c_k & s_k \\ s_k & -c_k \end{bmatrix} = \begin{bmatrix} \gamma_{k-1} & 0 \\ \delta_k & \bar{\gamma}_k \\ \varepsilon_{k+1} & \bar{\delta}_{k+1} \end{bmatrix},$$

with $\bar{\gamma}_1 = 1$, $\bar{\delta}_2 = t_2$, $c_k = \frac{\bar{\gamma}_{k-1}}{\gamma_{k-1}}$, and $s_k = \frac{t_k}{\gamma_{k-1}}$.

We show that $\delta_j = 0$ for all j by showing that $\bar{\gamma}_k = \frac{(-1)^k}{c_k}$ for $k \geqslant 2$, because in that case

$$\begin{aligned} \delta_k = \bar{\delta}_k c_k - (-1)^{k-1} s_k &= (t_k c_{k-1}) \frac{\bar{\gamma}_{k-1}}{\gamma_{k-1}} - (-1)^{k-1} \frac{t_k}{\gamma_{k-1}} \\ &= \frac{t_k}{\gamma_{k-1}} \left((-1)^{k-1} - (-1)^{k-1} \right) = 0. \end{aligned}$$

For $k = 2$ we have $\gamma_2^2 = 1 + t_2^2$ and $c_2 = \frac{1}{\gamma_2}$, so that $\bar{\gamma}_2 = \bar{\delta}_2 s_2 + c_2 = \frac{t_2^2}{\gamma_2} + \frac{1}{\gamma_2} = \gamma_2 = \frac{1}{c_2}$. Proceeding by induction, assume $c_{k-1} = \frac{(-1)^{k-1}}{\bar{\gamma}_{k-1}}$. Then

$$\begin{aligned} \bar{\gamma}_k = \bar{\delta}_k s_k - (-1)^{k-1} c_k &= \tfrac{1}{c_k} \left(-t_k c_{k-1} s_k c_k - (-1)^{k-1} c_k^2 \right) \\ &= -\tfrac{1}{c_k} \left((-1)^{k-1} \tfrac{t_k}{\bar{\gamma}_{k-1}} s_k c_k + (-1)^{k-1} c_k^2 \right) \\ &= \tfrac{(-1)^k}{c_k} \left(\frac{s_k}{c_k} s_k c_k + c_k^2 \right) = \tfrac{(-1)^k}{c_k}. \end{aligned}$$

For all k, since $\delta_k = 0$ and $x_k^{LQ} = W_{k-1} z_{k-1}$ with W_{k-1} having orthonormal columns, and since $(z_{k-1})_j = \zeta_j$ is defined by $L_{k-1} z_{k-1} = \|b\| e_1$, we have $\zeta_k = 0$ for k even. Therefore $x_{2k}^{LQ} = x_{2k-1}^{LQ}$. Furthermore, since $\zeta_k = c_k \bar{\zeta}_k$ and $x_k^{CG} = x_k^{LQ} + \bar{\zeta}_k \bar{w}_k$ for some $\bar{w}_k \perp W_k$, we have $\zeta_{2k} = 0$ and $x_{2k}^{CG} = x_{2k}^{LQ}$. The identity $x_{2k}^{CG} = x_k^C$ follows from (Saunders, 1995, Result 11). ◻

We illustrate Theorem 2 using a small numerical example. We randomly generate A and b with $m = 50$, $n = 30$, $M = I$, and $N = I$ and run SYMMLQ directly on (47). We compute $x_\star$ via Matlab's backslash operator, and compute $\|x_k - x_\star\|$ at each iteration to produce Figure 6. The resulting convergence plot resembles a staircase because every odd iteration produces a zero step.

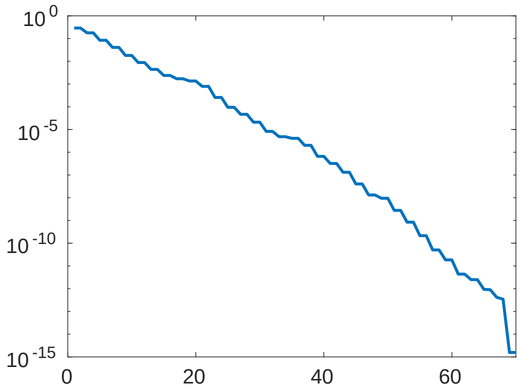

FIG. 6. *Error $\|x_k - x_\star\|$ generated by SYMMLQ applied to* (47). *Note that every odd iteration makes no progress, resulting in a convergence plot resembling a step function.*

9. Discussion. LNLQ fills a gap in the family of iterative methods for (3) based on the Golub and Kahan (1965) process. While CRAIG is equivalent to CG applied to $A^TAy = b$ (3), LNLQ is equivalent to SYMMLQ but is numerically more stable when A is ill-conditioned. The third possibility, MINRES (Paige and Saunders, 1975) applied to (3), is equivalent to LSQR (Paige and Saunders, 1982a,b) because both minimize the residual norm $\|b - Ax_k\|$, where $x_k \in \mathcal{K}_k$ is implicitly defined as $A^T y_k$.

As in the companion method LSLQ (Estrin et al., 2019b), an appropriate Gauss-Radau quadrature yields an upper bound on $\|y_k^L - y_\star\|$, and transition to the CRAIG point provides an upper bound on $\|y_k^C - y_\star\|$. However, it is x_k^C that is updated along orthogonal directions, and not x_k^L. Thus the upper bound on $\|x_k^L - x_\star\|$, which we developed for completeness, is deduced from that on $\|x_k^C - x_\star\|$. In our numerical experiments, both error bounds are remarkably tight, but $\|x_k^L - x_\star\|$ may lag behind $\|x_k^C - x_\star\|$ by several orders of magnitude and is not monotonic. Although the bound on $\|y_k^C - y_\star\|$ suggested by Arioli (2013) is tighter than might have been anticipated, the sliding window strategy allows us to tighten it further at the expense of a few extra scalar operations per iteration.

All error upper bounds mentioned above depend on an appropriate Gauss-Radau quadrature, which has been observed to become numerically inaccurate below a certain error level (Meurant and Tichý, 2014; Meurant and Tichý, 2018). This inaccuracy causes the loosening of the bounds observed in section 7. Should a more accurate computation of quadratic forms like $\|y_\star\|^2 = b^T(AA^T)^{-2}b$ become available, all error upper bounds would improve, including those from the sliding window approach.

USYMLQ, based on the orthogonal tridiagonalization process of Saunders, Simon, and Yip (1988), coincides with SYMMLQ when applied to consistent symmetric systems. For (3) it also coincides with LNLQ, but it would be wasteful to apply USYMLQ directly to (3).

Fong and Saunders (2012, Table 5.1) summarize the monotonicity of various quantities related to LSQR and LSMR iterations. Table 1 is similar but focuses on CRAIG and LNLQ.

Acknowledgements. We are grateful to Drew Kouri for the Matlab implementation of the PDE-constrained optimization problems used in the numerical experiments, and to the referees for their invaluable advice.

TABLE 1
Comparison of CRAIG and LNLQ properties on $\min \|x\|^2$ subject to $Ax = b$.

	CRAIG	LNLQ
$\|x_k\|$	$\nearrow$ (14) and (P, 1974)	non-monotonic
$\|x_\star - x_k\|$	$\searrow$ (12) and (P, 1974)	non-monotonic, $\geqslant$ CRAIG (Corollary 1)
$\|y_k\|$	$\nearrow$ (23) and (HS, 1952)	$\nearrow$ (23) and (PS, 1975), $\leqslant$ CRAIG (EOS, 2019)
$\|y_\star - y_k\|$	$\searrow$ (23) and (HS, 1952)	$\searrow$ (23) and (PS, 1975), $\geqslant$ CRAIG (EOS, 2019)
$\|r_\star - r_k\|$	not-monotonic	not-monotonic
$\|r_k\|$	not-monotonic	not-monotonic

$\nearrow$ monotonically increasing $\searrow$ monotonically decreasing
EOS (Estrin et al., 2019a), HS (Hestenes and Stiefel, 1952),
P (Paige, 1974), PS (Paige and Saunders, 1975)

References.

M. Arioli. Generalized Golub-Kahan bidiagonalization and stopping criteria. *SIAM J. Matrix Anal. Appl.*, 34(2):571–592, 2013. DOI: 10.1137/120866543.

J. E. Craig. The N-step iteration procedures. *J. Math. and Physics*, 34(1):64–73, 1955.

T. A. Davis and Y. Hu. The University of Florida sparse matrix collection. *ACM Trans. Math. Software*, 38(1):1:1–1:25, December 2011. DOI: 10.1145/2049662.2049663.

R. Estrin, M. P. Friedlander, D. Orban, and M. A. Saunders. Implementing a smooth exact penalty function for nonlinear optimization. Cahier du GERAD G-2018-XX, GERAD, 2018. In preparation.

R. Estrin, D. Orban, and M. A. Saunders. Euclidean-norm error bounds for CG via SYMMLQ. *SIAM J. Matrix Anal. Appl.*, 40(1):235–253, 2019a. DOI: 10.1137/16M1094816.

R. Estrin, D. Orban, and M. A. Saunders. LSLQ: An iterative method for linear least-squares with an error minimization property. *SIAM J. Matrix Anal. Appl.*, 40 (1):254–275, 2019b. DOI: 10.1137/17M1113552.

R. Fletcher. A class of methods for nonlinear programming: III. Rates of convergence. In F. A. Lootsma, editor, *Numerical Methods for Nonlinear Optimization*. Academic Press, New York, 1973.

D. C.-L. Fong and M. A. Saunders. CG versus MINRES: An empirical comparison. *SQU Journal for Science*, 17(1):44–62, 2012.

G. H. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *SIAM J. Numer. Anal.*, 2(2):205–224, 1965. DOI: 10.1137/0702016.

G. H. Golub and G. Meurant. Matrices, moments and quadrature. In *Numerical analysis 1993 (Dundee, 1993)*, volume 303 of *Pitman Res. Notes Math. Ser.*, pages 105–156. Longman Sci. Tech., Harlow, 1994.

G. H. Golub and G. Meurant. *Matrices, Moments and Quadrature with Applications*. Princeton Series in Applied Mathematics. Princeton University Press, Princeton, NJ, 2010. ISBN 978-0-691-14341-5.

G. H. Golub and Z. Strakŏs. Estimates in quadratic formulas. *Numer. Algor.*, 8(2-4): 241–268, 1994. DOI: 10.1007/BF02142693.

M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Standards*, 49(6):409–436, 1952.

W. J. Kammerer and M. Z. Nashed. On the convergence of the conjugate gradient method for singular linear operator equations. *SIAM J. Numer. Anal.*, 9(1):165–181, 1972. DOI: 10.1137/0709016.

C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Nat. Bur. Standards*, 45:225–280, 1950.

G. Meurant and P. Tichý. A new algorithm for computing quadrature-based bounds in conjugate gradients, 2014. URL http://www.cs.cas.cz/tichy/download/present/2014Spa.pdf.

G. Meurant and P. Tichý. Approximating the extreme ritz values and upper bounds for the A-norm of the error in CG. *Numerical Algorithms*, pages 1–32, 2018.

D. Orban and M. Arioli. *Iterative Solution of Symmetric Quasi-Definite Linear Systems*, volume 3 of *Spotlights*. SIAM, Philadelphia, 2017. DOI: 10.1137/1.9781611974737. URL http://bookstore.siam.org/sl03.

C. C. Paige. Bidiagonalization of matrices and solution of linear equations. *SIAM J. Numer. Anal.*, 11(1):197–209, 1974. DOI: 10.1137/0711019.

C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.*, 12(4):617–629, 1975. DOI: 10.1137/0712047.

C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Software*, 8(1):43–71, 1982a. DOI: 10.1145/355984.355989.

C. C. Paige and M. A. Saunders. Algorithm 583; LSQR: Sparse linear equations and least-squares problems. *ACM Trans. Math. Software*, 8(2):195–209, 1982b. DOI: 10.1145/355993.356000.

M. A. Saunders. Solution of sparse rectangular systems using LSQR and CRAIG. *BIT Numerical Mathematics*, 35:588–604, 1995. DOI: 10.1007/BF01739829.

M. A. Saunders, H. D. Simon, and E. L. Yip. Two conjugate-gradient-type methods for unsymmetric linear equations. *SIAM J. Numer. Anal.*, 25(4):927–940, 1988. DOI: 10.1137/0725052.

G. W. Stewart. The QLP approximation to the singular value decomposition. *SIAM J. Sci. Comput.*, 20(4):1336–1348, 1999. DOI: 10.1137/S1064827597319519.

R. J. Vanderbei. Symmetric quasi-definite matrices. *SIAM J. Optim.*, 5(1):100–113, 1995. DOI: 10.1137/0805005.