Les Cahiers du GERAD ISSN: 0711–2440

Estimates of the 2-norm forward error for SYMMLQ and CG

R. Estrin, D. Orban,
M.A. Saunders

G–2016–70

September 2016

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Estimates of the 2-norm forward error for SYMMLQ and CG

Ron Estrin [a]

Dominique Orban [b, c]

Michael A. Saunders [d]

[a] *Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA*

[b] *GERAD, Montréal (Québec) H3T 2A7, Canada*

[c] *Mathematics and Industrial Engineering Department, Polytechnique Montréal, Montréal (Québec) H3C 3A7, Canada*

[d] *Systems Optimization Laboratory, Department of Management Science and Engineering, Stanford University, Stanford, CA*

restrin@stanford.edu
dominique.orban@gerad.ca
saunders@stanford.edu

September 2016

Les Cahiers du GERAD

G–2016–70

Abstract: For positive definite linear systems (or semidefinite consistent systems), we use Gauss-Radau quadrature to obtain a cheaply computable upper bound on the 2-norm error of SYMMLQ iterates. The close relationship between SYMMLQ and CG iterates lets us construct an upper bound on the 2-norm error for CG. For indefinite systems, the upper bound becomes an estimate of the 2-norm SYMMLQ error. Numerical experiments demonstrate that the bounds and estimates are remarkably tight.

Keywords: Symmetric linear equations, iterative method, Krylov subspace method, Lanczos process, CG, SYMMLQ, error estimates

Résumé : La quadrature de Gauss-Radau nous permet d'obtenir une borne supérieure peu coûteuse sur l'erreur en norme Euclidienne associée aux itérés de SYMMLQ appliquées à un système symétrique et défini positif (ou un système semi-défini et consistent). La relation étroite entre les itérés de SYMMLQ et de CG fournit une borne supérieure sur l'erreur en norme Euclidienne associée à CG. Sur un système indéfini, la borne supérieure devient simplement une estimation de l'erreur en norme Euclidienne. Nos validations numériques montrent que les bornes et les estimations sont remarquablement proches de l'erreur exacte.

Mots clés : Systèmes d'équations linéaires symétriques, méthode itérative, méthode de Krylov, processus de Lanczos, CG, SYMMLQ, estimation de l'erreur

Acknowledgments: The research of D. Orban was partially supported by an NSERC Discovery Grant. M.A. Saunders's research was partially supported by the National Institute of General Medical Sciences of the National Institutes of Health [award U01GM102098].

1 Introduction

We consider the conjugate gradient method (CG) [9] and SYMMLQ [14] for solving symmetric linear systems $Ax = b$, where $A \in \mathbb{R}^{n\times n}$ is a sparse symmetric matrix or a fast linear operator. The kth iterates x_k^C and x_k^L formed by CG and SYMMLQ lie in the kth Krylov subspace

$$\mathcal{K}_k = \text{span}\left\{b, Ab, \ldots, A^{k-1}b\right\}.$$

In exact arithmetic, a property of Krylov methods ensures that there is an iteration $\ell \leqslant n$ such that the exact solution $x_\star = x_\ell^C = x_{\ell+1}^L$, where we note that x_k^L is defined for iterations $k = 2, \ldots, \ell + 1$.

When A is positive definite, it is known that the CG forward error $\|x_\star - x_k^C\|_2$ is monotonic [9, Thm 4:3], although it is not minimized in $\mathcal{K}_k$ at each iteration. The forward error is also monotonic for SYMMLQ, as it is minimized in a related space [16]. Empirically, CG typically maintains a smaller forward error than SYMMLQ by an order of magnitude, but neither CG nor SYMMLQ provide a way to estimate the error from above, although estimates of the CG forward error are developed by Golub and Meurant [7] and Meurant [11, 12]. Here, we derive cheaply computable estimates of the forward error for both methods. Our estimates are provably upper bounds when A is symmetric positive definite, and when A is symmetric positive semi-definite and the system is consistent.

Upper bounds and estimates of the forward error are desirable when the solution $x_\star$ has a meaningful interpretation, such as in parameter estimation. Although residual norms can be computed, they may give very loose error bounds that depend on the condition number of A.

In Section 2 we provide a brief overview of SYMMLQ. In Section 3 we derive upper bounds on the SYMMLQ and CG errors when A is positive semi-definite, and in Section 4 we discuss the implications when A is indefinite. In Section 5 we discuss possible ways of improving the estimates. In Section 6 we illustrate our results on problems from the UFL Sparse Matrix Collection. We discuss use of the error estimates in termination criteria in Section 7, and give concluding remarks in Section 8.

1.1 Notation

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

2 Overview of CG and SYMMLQ

There are many formulations of CG, but we focus on the following definition of x_k^C [9]:

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

Two equivalent formulations [16] of x_k^L are useful for our analysis:

$$\begin{aligned} x_k^L &= \underset{x\in\mathcal{K}_k}{\arg\min} \ \|x\| \ \text{ such that } \ b - Ax \perp \mathcal{K}_{k-1} \\ &= \underset{x\in A\mathcal{K}_{k-1}}{\arg\min} \ \|x_\star - x\|, \end{aligned}$$

where $A\mathcal{K}_{k-1} = \text{span}\left\{Ab, A^2b, \ldots, A^{k-1}b\right\}$.

Both CG and SYMMLQ may be derived from the Lanczos process [10], which generates orthonormal vectors $v_k \in \mathcal{K}_\ell$ such that, at the kth iteration, we have the factorization

$$AV_k = V_kT_k + \beta_{k+1}v_{k+1}e_k^T = V_{k+1}H_k,$$

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

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

With $\beta_1 = \|b\|$, the iterates $x_k^C = V_k y_k^C$ and $x_k^L = V_k y_k^L$ are defined by the subproblems

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

A more complete treatment of Krylov subspace methods for symmetric A can be found in [16].

2.1 The SYMMLQ iterates

We provide some key properties of SYMMLQ and describe some of the key quantities that are computed at the kth iteration. Many of the factorizations are reused and modified to obtain estimates of the SYMMLQ and CG error. A more detailed treatment can be found in [14], from which we derive most of the notation (with minor differences).

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

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

A single 2×2 reflection is applied to the factorization to obtain $H_{k-1}^T Q_k^T = [L_{k-1}\, 0]$, so that L_{k-1} differs from $\bar{L}_{k-1}$ only in the last diagonal entry, which becomes γ_{k-1}. The reflection is constructed so that

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

For $k \geqslant 2$, define $z_{k-1} = (\zeta_1, \dots, \zeta_{k-1})^T$ as the solution to $L_{k-1} z_{k-1} = \beta_1 e_1$. Note that $y_k^L = Q_k^T \begin{bmatrix} z_{k-1} \\ 0 \end{bmatrix}$, so that

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

with the orthogonal matrix $\overline{W}_k = V_k Q_k^T = [w_1, \dots, w_{k-1}, \bar{w}_k] = [W_{k-1}, \bar{w}_k]$.

The following results are proved in [14].

Lemma 1 *The SYMMLQ iterates x_k^L satisfy the following properties:*

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

3 Upper bounds on the forward error when A is SPD

In this section we assume A is SPD, and let $\lambda_{\text{est}} \in (0, \lambda_{\min}(A))$ be an underestimate of the smallest eigenvalue of A. We derive an upper bound on the forward error in SYMMLQ and build upon it to derive an upper bound for CG.

3.1 Existing error estimates for Krylov subspace methods

There has been significant interest in estimating the A-norm of the CG error, the history of which is detailed in [18]. The 2-norm has received less attention as it is more difficult to estimate for CG. It is studied primarily in [7, 11, 12], where, although estimates for the CG error are derived, they are not proved to be upper bounds. We are unaware of 2-norm SYMMLQ forward error estimates in the literature.

The strategy behind estimating error norms is to recognize the error and related quantities as quadratic forms $b^T f(A)b$ evaluated at A for a certain function f, and seek an upper bound on the value of this quadratic form. If $A = P\Lambda P^T$ is the eigenvalue decomposition of A, p_i is the ith column of P, and λ_i is the ith largest eigenvalue, then the quadratic form can be expressed as

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

The connection between such quadratic forms and their approximation via Gaussian quadrature is most notably studied by Golub and Meurant [6, 7], who show it is possible to derive upper and lower bounds by using the Lanczos decomposition of A. We follow this strategy to bound the SYMMLQ and CG errors.

3.2 Upper bounds on the SYMMLQ forward error

According to result 2 of Lemma 1 and (1), we have

$$\|x_\star - x_k^L\|^2 = \|x_\star\|^2 - \|x_k^L\|^2 = \|x_\star\|^2 - \|z_{k-1}\|^2. \tag{3}$$

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

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

We do not repeat the derivation of using Gauss-Radau quadrature to obtain an upper bound on such quadratic forms. The details can be found in [6, 8, 13]. The following key theorem is the basis of our approach.

Theorem 1 *Let A be SPD, $f : (0, \infty) \to \mathbb{R}$, and let the derivatives of f satisfy $f^{(2m+1)}(\xi) < 0$ for all $\xi \in (\lambda_{\min}(A), \lambda_{\max}(A))$ and all integers $m \geqslant 0$. Fix $\lambda_{\text{est}} \in (0, \lambda_{\min}(A))$. Let T_k be generated by k steps of the Lanczos process on (A, b) and let*

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

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

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

Proof. The result follows from [6, Theorem 3.2] and the section preceding it, as well as [6, Theorem 3.4]. □

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

The objective is to compute ω_k in $\widetilde{T}_k$, then efficiently evaluate the quadratic form. Golub and Meurant [6] show that $\omega_k = \lambda_{\text{est}} + \eta_{k-1}$, where η_{k-1} is obtained from the last entry of the solution of the system

$$(T_{k-1} - \lambda_{\text{est}} I)\, u_{k-1} = \beta_k^2 e_{k-1}. \tag{4}$$

To compute u_{k-1}, we take the QR factorization of $T_{k-1} - \lambda_{\text{est}} I$ in a similar way that we take the LQ factorization of H_{k-1}^T in SYMMLQ. The use of the QR factorization differs from [2], where a Cholesky factorization is used; this allows us to solve the system when A is indefinite using a stable factorization. The QR factorization begins with the 2×2 reflection

$$\begin{bmatrix} c_1^{(\omega)} & s_1^{(\omega)} \\ s_1^{(\omega)} & -c_1^{(\omega)} \end{bmatrix} \begin{bmatrix} \alpha_1 - \lambda_{\text{est}} & \beta_2 & \\ \beta_2 & \alpha_2 - \lambda_{\text{est}} & \beta_3 \end{bmatrix} = \begin{bmatrix} \rho_1 & \sigma_2 & \tau_3 \\ & \bar{\rho}_2 & \bar{\sigma}_3 \end{bmatrix},$$

and proceeds with reflections defined by

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

Putting the QR factorization together, we have

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

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

$$\begin{aligned} \bar{\rho}_1 &= \alpha_1 - \lambda_{\text{est}}, \\ \bar{\sigma}_2 &= \beta_2, \\ \rho_1 &= \sqrt{\bar{\rho}_1^2 + \beta_2^2}, \qquad c_1^{(\omega)} = \frac{\alpha_1 - \lambda_{\text{est}}}{\rho_1}, \qquad s_1^{(\omega)} = \frac{\beta_2}{\rho_1}; \end{aligned}$$

for $k \geqslant 2$:

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

From the QR factorization of Equation (4), we see that

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

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

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

$$\begin{aligned} \beta_1^2 e_1^T \widetilde{T}_k^{-2} e_1 &= \beta_1^2 e_1^T \left(\widetilde{L}_k \widetilde{Q}_k \right)^{-T} \left(\widetilde{L}_k \widetilde{Q}_k \right)^{-1} e_1 \\ &= \beta_1^2 e_1^T \widetilde{L}_k^{-T} \widetilde{L}_k^{-1} e_1 = \left\| \beta_1 \widetilde{L}_k^{-1} e_1 \right\|^2 \\ &= \left\| \tilde{z}_k \right\|^2, \end{aligned} \tag{5}$$

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

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

The vector $\widetilde{z}_k$ is closely related to z_k. Indeed $L_{k-1} z_{k-1} = \beta_1 e_1$, and therefore

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

Define what will be a bound on the forward error for the kth SYMMLQ iterate:

$$\epsilon_k^L := |\widetilde{\zeta}_k|. \tag{6}$$

Theorem 1 and (5) imply that $\|x_\star\|^2 \leqslant \|\widetilde{z}_k\|^2$ so that (3) yields

$$\|x_\star - x_k^L\|^2 = \|x_\star\|^2 - \|x_k^L\|^2 \leqslant \widetilde{\zeta}_k^2 = (\epsilon_k^L)^2.$$

Remarkably, with only a few extra floating-point operations we can compute an upper bound ϵ_k^L on the SYMMLQ 2-norm error.

Note that this approach can be applied when a positive definite preconditioner is used. The preconditioner changes the Lanczos decomposition, but all remaining computations carry through as above.

3.3 Upper bounds on the CG forward error

We now use the 2-norm error bound derived in the previous section to obtain an upper bound on the CG 2-norm error. We first establish that the CG error is always lower than that of SYMMLQ for A SPD. Although the result yields the trivial upper bound (6), it allows us to identify an improved bound. Define the kth CG direction as p_k with step length $\alpha_k^C > 0$, so that $x_k^C = \sum_{j=1}^k \alpha_j^C p_j$. First we recall key facts about the CG iterates.

Lemma 2 *The following results hold for CG:*

1. *The residuals $r_k^C = b - Ax_k^C$ are mutually orthogonal, and satisfy $r_k \perp \mathcal{K}_k$ [9, Theorem 5:1].*
2. *The search directions satisfy $p_i^T p_j \geqslant 0$ for all i, j [9, Theorem 5:2].*
3. *$\|x_k^C\|$ increases monotonically with k [17].*

The following lemma is also useful in our analysis.

Lemma 3 *For CG, $x_\star^T x_k^C \geqslant \|x_k^C\|^2$ for all $1 \leqslant k \leqslant \ell$.*

Proof. Lemma 2 yields

$$x_\star^T x_k^C = \left(x_k^C + \sum_{i=k+1}^{\ell} \alpha_i^C p_i \right)^T x_k^C = \left\| x_k^C \right\|^2 + \sum_{i=k+1}^{\ell} \sum_{j=1}^{k} \alpha_i^C \alpha_j^C p_i^T p_j \geqslant \left\| x_k^C \right\|^2, \tag{7}$$

because $\alpha_i^C > 0$ and $p_i^T p_j \geqslant 0$ for all i, j. □

We now relate the 2-norm errors of SYMMLQ and CG.

Theorem 2 *For CG and SYMMLQ on SPD $Ax = b$ with solution $x_\star$, the following hold in exact arithmetic for all $2 \leqslant k \leqslant \ell$:*

$$\|x_k^L\| \leqslant \|x_k^C\|, \tag{8}$$

$$\|x_\star - x_k^C\| \leqslant \|x_\star - x_k^L\|. \tag{9}$$

Proof. By definition, the SYMMLQ iterate x_k^L is optimal for the problem

$$\min_{x \in \mathcal{K}_k} \|x\| \text{ such that } r \perp \mathcal{K}_{k-1} \qquad (r = b - Ax). \tag{10}$$

Note that x_k^C is feasible for Equation (10) because $x_k^C \in \mathcal{K}_k$ and by Lemma 2, $r_k^C \perp \mathcal{K}_k$. Since x_k^L is optimal for Equation (10), we obtain (8). Lemma 3 and Equation (8) together imply

$$\|x_k^L\|^2 + \|x_k^C\|^2 \leqslant 2\|x_k^C\|^2 \leqslant 2x_\star^T x_k^C.$$

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

$$\|x_\star\|^2 - 2x_\star^T x_k^C + \|x_k^C\|^2 \leqslant \|x_\star\|^2 - \|x_k^L\|^2.$$

By factoring the left and using result 2 of Lemma 1 on the right, we obtain (9). □

Although the proof of Theorem 2 assumes exact arithmetic, we have observed empirically that the result holds until x_k^L approaches $x_\star$.

Theorem 2 immediately establishes the trivial bound

$$\|x_\star - x_k^C\| \leqslant \|x_\star - x_k^L\| \leqslant \epsilon_k^L, \tag{11}$$

which provides us with a provable upper bound on the 2-norm CG error, in contrast with the estimates in [12]. We can improve the naive bound using a few observations. Result 3 of Lemma 1 and Equation (8) yield $\bar{\zeta}_k^2 = \|x_k^C\|^2 - \|x_k^L\|^2 \geqslant 0$. From Lemma 3,

$$\theta_k := x_\star^T x_k^C - \|x_k^C\|^2 \geqslant 0. \tag{12}$$

Hence

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

and since $\|x_\star - x_k^L\| \leqslant \tilde{\zeta}_k$ it follows that

$$\begin{aligned}\|x_\star - x_k^C\|^2 &= \|x_\star - x_k^L\|^2 - \bar{\zeta}_k^2 - 2\theta_k \\ &\leqslant \tilde{\zeta}_k^2 - \bar{\zeta}_k^2 - 2\theta_k \qquad (13) \\ &\leqslant \tilde{\zeta}_k^2 - \bar{\zeta}_k^2. \qquad (14)\end{aligned}$$

Since $\bar{\zeta}_k$ is readily available as part of the SYMMLQ iteration, (14) is an improvement upon the bound Equation (11). Unfortunately since $x_\star$ is unavailable, the bound in Equation (13) is not computable. We define

$$\epsilon_k^C := \sqrt{\tilde{\zeta}_k^2 - \bar{\zeta}_k^2} \leqslant \left|\tilde{\zeta}_k\right| = \epsilon_k^L \tag{15}$$

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

In view of Equation (7), we can improve the error estimate by approximating θ_k from below using the sliding window approach used in [2, 4, 12]. Based on result 2 of Lemma 2, we define an approximation of Equation (12) as

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

noting that $0 \leqslant \theta_k^{(1)} \leqslant \cdots \leqslant \theta_k^{(\ell-k-1)} = \theta_k$. Therefore for modest values of d, the CG iterates $x_k^C, \ldots, x_{k+d}^C$ can be stored in order to compute $\theta_k^{(d)}$, which improves the bound in Equation (14) to

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

Since the sliding window is more memory intensive than the computation of ϵ_k^C, we focus the rest of the discussion on ϵ_k^C as an upper bound on the CG error.

For positive definite A, we have derived provable upper bounds on the error at every iteration of SYMMLQ and CG. These error bounds require only a few scalar operations per iteration, and only $O(1)$ extra memory. In Section 6 we evaluate how well they track the true error for various values of λ_{est}.

3.4 Estimation of $\|x_\star - x_k^L\|$ with A semidefinite

Suppose that $Ax = b$ is a singular but consistent system, where A is semidefinite with rank $r < n$. SYMMLQ and CG then find the pseudoinverse solution $x_\star = A^\dagger b = \arg\min_x \|x\|$ s.t. $Ax = b$. As before, in order to estimate the solution, we need to estimate the quadratic form $\|x_\star\|^2 = b^T \left(A^\dagger\right)^2 b = b^T f(A) b$, where

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

From the eigenvalue decomposition $A = P \Lambda P^T$, where p_i is the ith column of P and λ_i is the ith largest eigenvalue of A, this quadratic form is expressible as

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

Compared to Equation (2), the only difference is that we now compute the sum over the nonzero eigenvalues. If the spectrum of A is ordered as $0 = \lambda_n = \cdots = \lambda_{r+1} < \lambda_r \leqslant \cdots \leqslant \lambda_1$, the Rayleigh-Ritz theorem states that

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

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

We can therefore modify Theorem 1 to allow for $\xi \in (\lambda_r, \lambda_{\max}(A))$ rather than $\xi \in (\lambda_{\min}(A), \lambda_{\max}(A))$ without any changes to the derivation of the upper bounds. To compute upper bounds, we seek underestimates to the smallest nonzero eigenvalue rather than the smallest eigenvalue (which is zero in this case) so that $\lambda_{\text{est}} \in (0, \lambda_r)$.

4 Estimation of $\|x_\star - x_k^L\|$ with A indefinite

We now focus on the SYMMLQ error when A is indefinite. Theorem 1 no longer applies, and so $\beta_1^2 e_1^T \tilde{T}_k^{-2} e_1$ is only an estimate of $\|x_\star\|$ rather than an upper bound.

There are two approaches. The first is to continue as in Section 3.2 and accept ϵ_k^L as an estimate of the error rather than an upper bound. Alternatively we can treat $\|x_\star\|^2 = b^T A^{-2} b$ as a quadratic form in A^2 rather than A. We formulate the problem as upper bounding the energy norm $\|x_\star\| = \|b\|_{B^{-1}}$ with $B = A^2$. Such computation is akin to computing the energy norm error for CG using Gauss-Radau quadrature, which has been studied in works such as [7]. The main difficulty is that it requires applying the Lanczos process to A^2 and b, which means two applications of A per iteration of SYMMLQ. Although this theoretically guarantees that we obtain an upper bound on $\|x_\star\|$ (and therefore an upper bound on the error), roundoff error can diminish the quality of the estimation.

With these ideas in mind, we consider using the procedure outlined in Section 3.2, treating $b^T A^{-2} b$ as a quadratic form in A to estimate the forward error. In numerical experiments we observe that the estimate often remains an upper bound, even as the iterates converge to the solution. Furthermore, it is possible to loosen the error estimate by choosing a smaller value for λ_{est} to encourage the estimate to remain an upper bound. This is also illustrated in the numerical experiments.

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

5 Choosing λ_{est}

We make a few comments about the choice of λ_{est}. First, if the smallest eigenvalue $\lambda_{\min} = \arg\min_{\lambda \in \lambda(A)} |\lambda|$, is known exactly, one should choose $\lambda_{\text{est}} = (1 - \epsilon)\lambda_{\min}$ with $\epsilon \ll 1$. In the numerical experiments below, we choose $\epsilon = 10^{-10}$. Choosing λ_{est} slightly smaller than the minimum eigenvalue alleviates numerical stability issues in computing ω_k with a near-singular $T_k - \lambda_{\text{est}} I$. This also applies when A is indefinite.

When $\lambda_{\min}$ is not known, the choice of λ_{est} becomes application-specific. Another consideration is shifted linear systems $(A+\delta I)x = b$ with A SPD and $\delta > 0$, where the choice $\lambda_{\text{est}} = \delta$ may give good error estimates if A is close to singularity. This is of interest for regularized least-squares problems [4, 15] and is exploited in [4].

6 Numerical experiments

We evaluate the quality of the error bounds on some examples taken from the UFL Sparse Matrix Collection [3]. We use a Matlab implementation of SYMMLQ with error bounds added as described in Section 3. For each experiment, we solve $Ax = b$ with a random b of unit norm, and take $x_\star$ to be the output of Matlab's backslash operator $A\backslash b$. For both SYMMLQ and CG we use the termination criterion $\|r_k\| / \|b\| \leqslant 10^{-10}$. We compute $\lambda_{\min}(A)$, the eigenvalue closest to zero, and obtain the error bounds using $\lambda_{\text{est}} = (1-10^{-10})\lambda_{\min}(A)$ and $\lambda_{\text{est}} = \frac{1}{10}\lambda_{\min}(A)$. We also include a lower-bound error estimate using the sliding-window approach [9]. Since SYMMLQ takes orthogonal steps,

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

for any $d \geqslant 1$. Thus by choosing a modest value $d = 5$ or 10 and storing the last d steplengths ζ_i, we can compute a lower bound on the error. We note that it's also possible to compute a lower bound via Gauss and Gauss-Radau quadrature with $\lambda_{\text{est}} \geqslant \|A\|_2$. Such techniques were used in [1], and they provide comparable lower bounds to those from the sliding window.

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

We first consider SYMMLQ on examples where A is SPD. For HB/bcsstk28 with $n = 4410$ and $\kappa(A) \approx 10^8$, the errors and upper bounds are shown in Figure 1. For UTEP/Dubcova1 with $n = 16129$ and $\kappa(A) \approx 10^3$, they are in Figure 2. We see that when λ_{est} approximates $\lambda_{\min}$ well, the bound ϵ_k^L is remarkably tight after an initial lag. Even when λ_{est} is a tenth of the true eigenvalue, it appears that the bound is at most one order of magnitude larger, still outlining the true error from above. Only near convergence, ϵ_k^L may no longer be a bound as the true error plateaus. Having the computed bound continue to decrease after convergence is a desirable property for termination criteria. The lower bounds $\epsilon_k^L(5)$ and $\epsilon_k^L(10)$ appear quite loose until reasonable progress begins in Figure 1, but this is not an issue in Figure 2, where both the upper and lower bounds approximate the true error well.

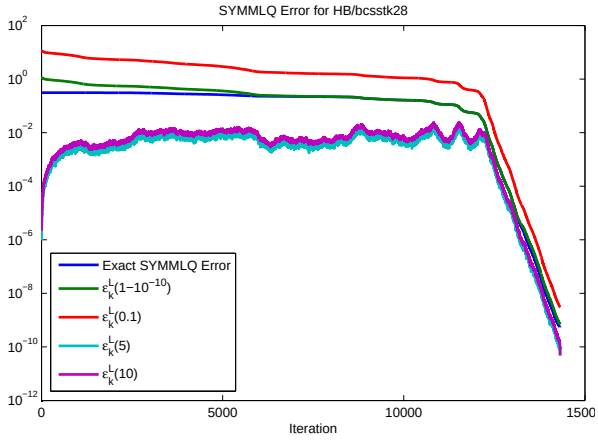

Figure 1: $\|x_\star - x_k^L\|$ and error bounds for SPD system HB/bcsstk28. The Gauss-Radau approach gives upper bounds, while the sliding window approach gives lower bounds.

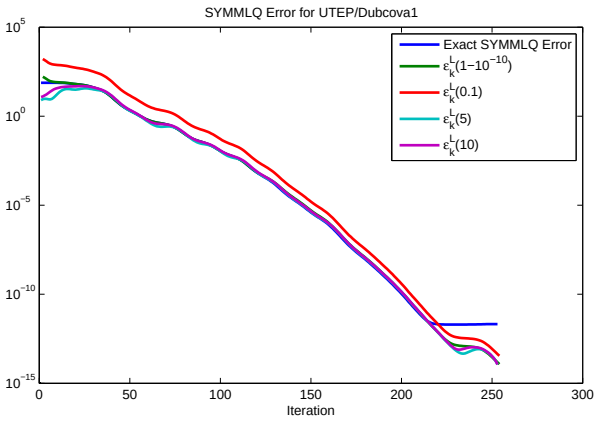

Figure 2: $\|x_\star - x_k^L\|$ and error bounds for SPD system UTEP/Dubcova1. The Gauss-Radau approach gives upper bounds, while the sliding window approach gives lower bounds.

We now solve the same problems using CG (via the SYMMLQ transfer point). Figures 3 and 4 show that ϵ_k^C is a considerably looser bound on the CG error than ϵ_k^L is on the SYMMLQ error, although they remain true upper bounds until convergence. As with SYMMLQ, if the error stagnates at convergence, the "bound" may continue to decrease. Also when the error is $O(10^{-4})$, ϵ_k^C diverges slightly from the true CG error. This is probably due to $\bar{\zeta}_k$ becoming an order of magnitude smaller than ϵ_k^L.

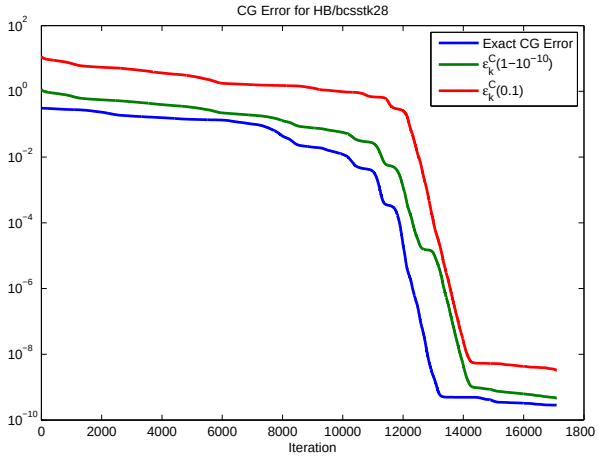

Figure 3: $\|x_\star - x_k^C\|$ and error bounds for SPD system HB/bcsstk28.

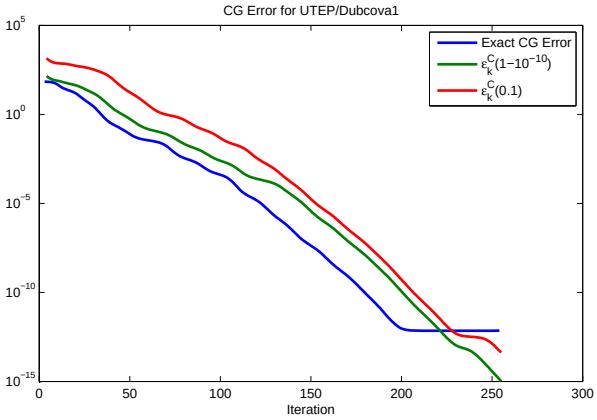

Figure 4: $\|x_\star - x_k^C\|$ and error bounds for SPD system UTEP/Dubcova.

Next we consider examples where A is indefinite, using matrices PARSEC/Na5 and PARSEC/SiNa with $n = 5822$ and 5743 and $\kappa(A) \approx 10^3$ and 10^2. Figure 5 shows that the error estimate based on quadrature is not an upper bound for all iterations, and sometimes dips below the sliding window lower-bound. Interestingly, as the iterates converge, the quadrature estimate becomes an upper bound again. We were not able to establish such a property, but it is an empirical observation on many problems. We can encourage ϵ_k^L to remain an upper bound by underestimating the magnitude of $\lambda_{\min}$, although this is again heuristic. In Figure 6, even though A is indefinite, we see that the error estimate using $\lambda_{\min}$ remains an upper bound (until convergence) and closely approximates the true error. Underestimation of $\lambda_{\min}$ again loosens the bound, but the approximation remains quite close.

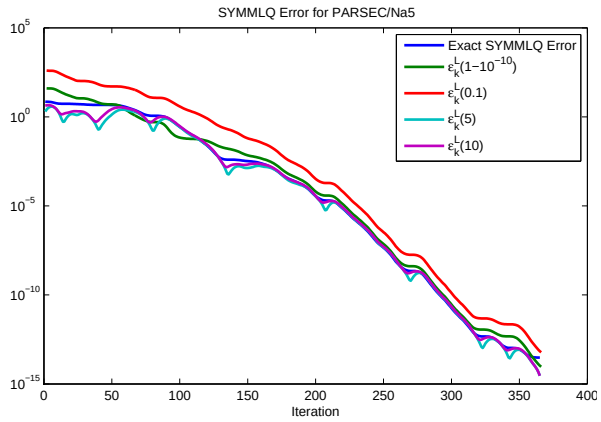

Figure 5: $\|x_\star - x_k^L\|$ and error estimates for indefinite system PARSEC/Na5. The Gauss-Radau approach no longer guarantees an upper bound, but tends to track from above after an initial lag. The sliding window approach continues to provide a lower bound.

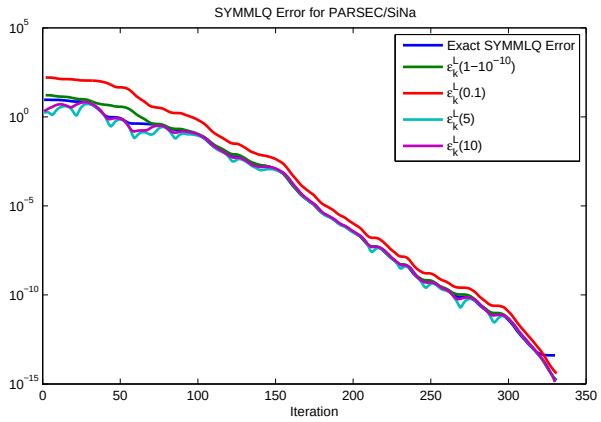

Figure 6: $\|x_\star - x_k^L\|$ and error estimates for indefinite system PARSEC/SiNa. The Gauss-Radau approach no longer guarantees an upper bound, but in this case provides an upper bound anyway. The sliding window approach continues to provide a lower bound.

7 2-norm-based termination criteria

We make a few comments about the potential use of ϵ_k^C and ϵ_k^L in termination criteria for CG and SYMMLQ. For SPD systems, we have seen in practice that if λ_{est} is close to $\lambda_{\min}$, the error bounds are remarkably tight. When λ_{est} is loose, we observe that $|\lambda_{\min}(A)|/|\lambda_{\text{est}}| \approx \epsilon_k^L/\|x_\star - x_k^L\|$. It was shown in Section 6 that the error estimate is an upper bound until convergence, after which the true error may plateau but ϵ_k^C and ϵ_k^L continue to decrease. This property makes it possible to terminate the iterations as soon as ϵ_k^L or ϵ_k^C drops below a prescribed level.

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

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

8 Concluding remarks

We have developed a cheap estimate for the forward error along the SYMMLQ and CG iterations, and explored the relationship between the forward errors of the two methods. Fong and Saunders [5, Table 5.1] summarize the monotonicity of various quantities related to the CG and MINRES iterations. Table 1 repeats the information and adds SYMMLQ for a more complete picture.

When A is SPD, our error estimate is proven to be an upper bound prior to convergence. For CG, the estimate can be made tighter by combining it with a sliding window approximation of the difference between the true error and estimated error at the expense of storing a constant number of extra vectors during the iterations. When A is indefinite, the estimate is not guaranteed to be an upper bound, but often tracks the error closely after an initial lag.

Table 1: Comparison of CG, MINRES, and SYMMLQ properties on an SPD system $Ax = b$. Italicized results hold for indefinite systems as well.

	CG	MINRES	SYMMLQ
$\|x_k\|$	↗ [17, Thm 2.1]	↗ [5, Thm 2.3]	↗ *[14]*, ⩽ CG (Thm 2)
$\|x_\star - x_k\|$	↘ [9, Thm 4:3]	↘ [9, Thm 7:5]	↘ *[14]*, ⩾ CG (Thm 2)
$\|x_\star - x_k\|_A$	↘ [9, Thm 6:3]	↘ [9, Thm 7:4]	not-monotonic
$\|r_k\|$	not-monotonic	↘ *[9, Thm 7:2]*	not-monotonic
$\|r_k\| / \|x_k\|$	not-monotonic	↘ [5, Thm 3.1]	not-monotonic
↗ monotonically increasing		↘ monotonically decreasing	

References

[1] M. Arioli. Generalized Golub-Kahan bidiagonalization and stopping criteria. SIAM J. Matrix Anal. Appl., 34(2):571–592, 2013.

[2] M. Arioli and D. Orban. Iterative methods for symmetric quasi-definite linear systems. Part I: Theory. Technical report, GERAD, 2013.

[3] T. A. Davis and Y. Hu. The University of Florida sparse matrix collection. ACM Trans. Math. Softw., 38(1):1:1–1:25, December 2011.

[4] R. Estrin, D. Orban, and M. A. Saunders. LSLQ: An iterative method for linear least-squares problems with a forward error minimization property. Technical report, GERAD, 2016. In preparation.

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

[6] 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.

[7] G. H. Golub and G. Meurant. Matrices, moments and quadrature II; how to compute the norm of the error in iterative methods. BIT Numerical Mathematics, 37(3):687–705, 1997.

[8] G. H. Golub and G. Meurant. Matrices, Moments and Quadrature with Applications. Princeton University Press, Princeton, NJ, 2009.

[9] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. J. Res. Nat. Bureau Standards, 49:409–436, 1952.

[10] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. J. Res. Nat. Bureau Standards, 45(4):255–282, October 1950.

[11] G. Meurant. The computation of bounds for the norm of the error in the conjugate gradient algorithm. Numerical Algorithms, 16(1):77–87, 1997.

[12] G. Meurant. Estimates of the l2 norm of the error in the conjugate gradient algorithm. Numerical Algorithms, 40(2):157–169, 2005.

[13] G. Meurant. The Lanczos and Conjugate Gradient Algorithms: From Theory to Finite Precision Computations (Software, Environments, and Tools). SIAM, 2006.

[14] C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. SIAM J. Numer. Anal., 12(4):617–629, 1975.

[15] C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. ACM Trans. Math. Softw., 8(1):43–71, March 1982.

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

[17] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. SIAM J. Numer. Anal., 20(3):626–637, 1983.

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