

MINIMAL RESIDUAL METHODS FOR COMPLEX SYMMETRIC, SKEW SYMMETRIC, AND SKEW HERMITIAN SYSTEMS*

SOU-CHENG T. CHOI†

Dedicated to Michael Saunders's 70th birthday

Abstract. While there is no lack of efficient Krylov subspace solvers for Hermitian systems, there are few for complex symmetric, skew symmetric, or skew Hermitian systems, which are increasingly important in modern applications including quantum dynamics, electromagnetics, and power systems. For a large consistent complex symmetric system, one may apply a non-Hermitian Krylov subspace method disregarding the symmetry of A , or a Hermitian Krylov solver on the equivalent normal equation or an augmented system twice the original dimension. These have the disadvantages of increasing either memory, conditioning, or computational costs. An exception is a special version of QMR by Freund (1992), but that may be affected by non-benign breakdowns unless look-ahead is implemented; furthermore, it is designed for only consistent and nonsingular problems. For skew symmetric systems, Greif and Varah (2009) adapted CG for nonsingular skew symmetric linear systems that are necessarily and restrictively of even order.

We extend the symmetric and Hermitian algorithms MINRES and MINRES-QLP by Choi, Paige and Saunders (2011) to complex symmetric, skew symmetric, and skew Hermitian systems. In particular, MINRES-QLP uses a rank-revealing QLP decomposition of the tridiagonal matrix from a three-term recurrent complex-symmetric Lanczos process. Whether the systems are real or complex, singular or invertible, compatible or inconsistent, MINRES-QLP computes the unique minimum-length, i.e., pseudoinverse, solutions. It is a significant extension of MINRES by Paige and Saunders (1975) with enhanced stability and capability.

Key words. MINRES, MINRES-QLP, Krylov subspace method, Lanczos process, conjugate-gradient method, minimum-residual method, singular least-squares problem, sparse matrix, complex symmetric, skew symmetric, skew Hermitian, preconditioner, structured matrices

AMS subject classifications. 15A06, 65F10, 65F20, 65F22, 65F25, 65F35, 65F50, 93E24

DOI. xxx/xxxxxxxxx

1. Introduction. Krylov subspace methods are generally divided into two classes: Those for Hermitian matrices (e.g. CG [27], MINRES [38], SYMMLQ [38], MINRES-QLP [8, 9, 11, 7]) and those for general matrices without such symmetries (e.g. BiCG [15], GMRES [41], QMR [19], BiCGstab [52], LSQR [39, 40]). Such a division is largely due to historical reasons in numerical linear algebra—the most prevalent structure for matrices arising from practical applications is that of being Hermitian (which reduces to symmetric for real matrices). However other types of symmetry structures, notably complex symmetric, skew-symmetric, and skew-Hermitian matrices, are becoming increasingly common in modern applications. Currently, aside possibly for storage and matrix-vector products, these are treated like any general matrices with no symmetry structures. The algorithms in this article go substantially further in developing specialized Krylov subspace algorithms designed at the outset to exploit these symmetry structures. In addition, our algorithms constructively reveal the (numerical) compatibility and singularity of a given linear system—users do not have to know these properties a priori.

*Draft CSMINRES18.tex of April 26, 2013.

†Computation Institute, University of Chicago, Chicago, IL 60637 (sctchoi@uchicago.edu). This work was supported in part by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Dept. of Energy, under Contract DE-AC02-06CH11357 and by the National Science Foundation grant SES-0951576 through the Center for Robust Decision Making on Climate and Energy Policy.

We are concerned with iterative methods for solving a large linear system $Ax = b$ or the more general minimum-length least-squares (LS) problem

$$\min \|x\|_2 \quad \text{s.t.} \quad x \in \arg \min_{x \in \mathbb{C}^n} \|Ax - b\|_2, \quad (1.1)$$

where $A \in \mathbb{C}^{n \times n}$ is complex symmetric ($A = A^T$) or skew Hermitian ($A = -A^*$), and possibly singular; and $b \in \mathbb{C}^n$. Our results are directly applicable to problems with symmetric or skew symmetric matrices $A = \pm A^T \in \mathbb{R}^{n \times n}$ and real vectors b . A may exist only as an operator for returning the product Ax .

The solution of (1.1), called the *minimum-length* or *pseudoinverse* solution [21], is formally given by $x^\dagger = (A^*A)^\dagger A^*b$, where A^\dagger denotes the pseudoinverse of A . The pseudoinverse is continuous under perturbations E for which $\text{rank}(A + E) = \text{rank}(A)$ [46], and x^\dagger is continuous under the same condition. Problem (1.1) is then well-posed [24].

Let $A = U\Sigma U^T$ be a Takagi decomposition [29], a singular-value decomposition (SVD) specialized for a complex symmetric matrix, with U unitary ($U^*U = I$) and $\Sigma \equiv \text{diag}([\sigma_1, \dots, \sigma_n])$ real non-negative and $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$, where r is the rank of A . We define the condition number of A to be $\kappa(A) = \frac{\sigma_1}{\sigma_r}$, and we say that A is ill-conditioned if $\kappa(A) \gg 1$. Hence a mathematically nonsingular matrix (e.g., $A = \begin{bmatrix} 1 & 0 \\ 0 & \varepsilon \end{bmatrix}$, where ε is the machine precision) could be regarded as numerically singular. Also, a singular matrix could be well-conditioned or ill-conditioned. For a skew Hermitian matrix, we use its (full) eigenvalue decomposition $A = V\Lambda V^*$, where Λ is a diagonal matrix of imaginary numbers (possibly zeros; in conjugate pairs if A is real, i.e., skew symmetric) and V is unitary¹. We define its condition number as $\kappa(A) = \frac{|\lambda_1|}{|\lambda_r|}$, the ratio of the largest and smallest nonzero eigenvalues in magnitude.

EXAMPLE 1.1. *We contrast the five classes of symmetric or Hermitian matrices by their definitions and small instances of order $n = 2$:*

$$\mathbb{R}^{n \times n} \ni A = A^T = \begin{bmatrix} 1 & 5 \\ 5 & 1 \end{bmatrix} \text{ is symmetric.}$$

$$\mathbb{C}^{n \times n} \ni A = A^* = \begin{bmatrix} 1 & 1 - 2i \\ 1 + 2i & 1 \end{bmatrix} \text{ is Hermitian (with real diagonal).}$$

$$\mathbb{C}^{n \times n} \ni A = A^T = \begin{bmatrix} 2 + i & 1 - 2i \\ 1 - 2i & i \end{bmatrix} \text{ is complex symmetric (with complex diagonal).}$$

$$\mathbb{R}^{n \times n} \ni A = -A^T = \begin{bmatrix} 0 & 5 \\ -5 & 0 \end{bmatrix} \text{ is skew symmetric (with zero diagonal).}$$

$$\mathbb{C}^{n \times n} \ni A = -A^* = \begin{bmatrix} 0 & 1 - 2i \\ -1 - 2i & 0 \end{bmatrix} \text{ is skew Hermitian (with zero diagonal).}$$

CG, SYMMLQ, and MINRES are designed for solving nonsingular symmetric systems $Ax = b$. CG is efficient on symmetric positive definite systems. For indefinite problems, SYMMLQ and MINRES are reliable even if A is ill-conditioned.

Choi [7] appeared to be the first work that comparatively analyzed the algorithms on singular symmetric and Hermitian problems. On (singular) incompatible problems

¹Skew Hermitian (symmetric) matrices are, like Hermitian matrices, unitarily diagonalizable (i.e., normal [51, Theorem 24.8]).

CG and SYMMLQ iterates x_k diverge to some nullvectors of A [7, Propositions 2.7, 2.8, and 2.15; Lemma 2.17]. MINRES often seems more desirable to users because its residual norms are monotonically decreasing. On singular compatible systems, MINRES returns x^\dagger [7, Theorem 2.25]. On singular incompatible systems, MINRES remains reliable if it is terminated with a suitable stopping rule that monitors $\|Ar_k\|$ [8, Lemma 3.3], but the solution is generally not x^\dagger [8, Theorem 3.2]. MINRES-QLP [8, 9, 11, 7] is a significant extension of MINRES, capable of computing x^\dagger , simultaneously minimizing residual and solution norms. The additional cost of MINRES-QLP is moderate relative to MINRES: 1 vector in memory, 4 axpy ($y \leftarrow \alpha x + y$), and 3 vector scaling ($x \leftarrow \alpha x$) per iteration. The efficiency of MINRES is partially, and in some cases almost fully, retained in MINRES-QLP by transferring from a *MINRES phase* to a *MINRES-QLP phase* only when an estimated $\kappa(A)$ exceeds a user-specified value. The MINRES phase is optional, consisting of only MINRES iterations for nonsingular and well-conditioned subproblems. The MINRES-QLP phase handles less well-conditioned and possibly numerically singular subproblems. In all iterations, MINRES-QLP uses QR factors of the tridiagonal matrix from a Lanczos process and then applies a second QR decomposition on the conjugate transpose of the upper-triangular factor to obtain and reveal the rank of a lower-tridiagonal form. On nonsingular systems, MINRES-QLP enhances the accuracy (with less rounding errors) and stability of MINRES. It is applicable to symmetric and Hermitian problems with no traditional restrictions such as nonsingularity and definiteness of A or compatibility of b .

The aforementioned established Hermitian methods are not directly applicable to complex or skew symmetric equations. For consistent complex symmetric problems, which could arise in Helmholtz equations, linear systems that involve Hankel matrices, or applications in quantum dynamics, electromagnetics, and power systems, we may apply a non-Hermitian Krylov subspace method disregarding the symmetry of A , or a Hermitian Krylov solver (such as CG, SYMMLQ, MINRES, or MINRES-QLP) on the equivalent normal equation or an augmented system twice the original dimension. They suffer increasing memory, conditioning, or computational costs. An exception² is a special version of QMR by Freund (1992) [18], which takes advantage of the matrix symmetry by using an unsymmetric Lanczos framework. Unfortunately, it is known that the algorithm may be affected by non-benign breakdowns unless a look-ahead strategy is implemented. Another less than elegant feature of QMR is the vector norm of choice is induced by the inner product $x^T y$ but it is not a proper vector norm (e.g., $0 \neq x^T := [1 \ i]$, where $i = \sqrt{-1}$, yet $x^T x = 0$). Besides, QMR is designed for only nonsingular and consistent problems. Inconsistent complex symmetric problems (1.1) could arise from shifted problems in inverse or Rayleigh quotient iterations; mathematically or numerically singular or inconsistent systems, in which A or b are vulnerable to errors due to measurement, discretization, truncation, or round-off. In fact, QMR and most non-Hermitian Krylov solvers (other than LSQR) fail to converge to x^\dagger on an example as simple as $A = i \operatorname{diag}(\begin{bmatrix} 1 \\ 0 \end{bmatrix})$ and $b = i \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, for which $x^\dagger = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$.

Here we extend the symmetric and Hermitian algorithms MINRES and MINRES-QLP to complex symmetric systems. The main aim is to deal reliably with compatible or incompatible systems and to return the *unique* solution of (1.1). Like QMR and the Hermitian Krylov solvers, it exploits the matrix symmetry.

Noting the similarities in the definitions of skew symmetric matrices ($A = -A^T \in \mathbb{R}^{n \times n}$) and complex symmetric matrices and motivated by algebraic Riccati equa-

²It is noteworthy that among direct methods for large sparse systems, MA57 and ME57 [14] are available for real and complex symmetric problems.

tions [32] and more recent, novel applications of Hodge theory in data-mining [33, 20], we evolve MINRES-QLP furthermore for solving skew symmetric linear systems. Greif and Varah [22] adapted CG for nonsingular skew symmetric linear systems that are *skew- A conjugate*, meaning $-A^2$ is symmetric positive definite. The algorithm is further restricted to A of even-order since a skew symmetric matrix of odd order is singular. Our MINRES-QLP extension has no such limitations and are applicable to singular problems. For skew Hermitian problems with skew Hermitian matrices or operators ($A = -A^* \in \mathbb{C}^{n \times n}$), our approach is to transform them into Hermitian systems so that they could immediately take advantage of the original Hermitian version of MINRES-QLP.

1.1. Notation. For an incompatible system, $Ax \approx b$ is shorthand for the LS problem (1.1). We use “ \simeq ” to mean “approximately equal to”. The letters i, j, k in subscripts or superscripts denote integer indices, otherwise they may represent $\sqrt{-1}$; c and s cosine and sine of some angle θ ; e_k the k th unit vector; e a vector of all ones; and other lower-case letters such as b, u , and x (possibly with integer subscripts) denote *column* vectors. Upper-case letters A, T_k, V_k, \dots denote matrices, and I_k is the identity matrix of order k . Lower-case Greek letters denote scalars; in particular, $\varepsilon \simeq 10^{-16}$ denotes the floating-point double precision. If a quantity δ_k is modified one or more times, we denote its values by $\delta_k, \delta_k^{(2)}, \dots$. We use $\text{diag}(v)$ to denote a diagonal matrix with elements of a vector v on the diagonal. The transpose, conjugate, and conjugate transpose of a matrix A is denoted as A^T, \bar{A} , and $A^* = \bar{A}^T$ respectively. The symbol $\|\cdot\|$ denotes the 2-norm of a vector ($\|x\| = \sqrt{x^*x}$) or a matrix ($\|A\| = \sigma_1$ from A ’s SVD).

1.2. Overview. In Section 2 we briefly review the Lanczos processes and QLP decomposition before developing the algorithms in Sections 3-5. Preconditioned algorithms are described in Section 6. Numerical experiments are described in Section 7. We conclude with future work and related software in the last section, Section 8. Our pseudocode and a summary of norm estimates and stopping conditions are given in Appendices A and B.

2. Review. In the following few subsections, we summarize algebraic methods necessary for our algorithmic development.

2.1. Lanczos processes. Given a complex symmetric operator A and a vector b , a Lanczos-like³ process [2, 23], which we name as the *Saunders process*, computes vectors v_k and tridiagonal matrices \underline{T}_k according to $v_0 \equiv 0$, $\beta_1 v_1 = b$, and then⁴

$$p_k = A\bar{v}_k, \quad \alpha_k = v_k^* p_k, \quad \beta_{k+1} v_{k+1} = p_k - \alpha_k v_k - \beta_k v_{k-1} \quad (2.1)$$

for $k = 1, 2, \dots, \ell$, where we choose $\beta_k > 0$ to give $\|v_k\| = 1$. In matrix form,

$$A\bar{V}_k = V_{k+1} \underline{T}_k, \quad \underline{T}_k \equiv \begin{bmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_k \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{bmatrix} \equiv \begin{bmatrix} T_k \\ \beta_{k+1} e_k^T \end{bmatrix}, \quad V_k \equiv [v_1 \quad \dots \quad v_k]. \quad (2.2)$$

³We distinguish our process from the complex symmetric Lanczos process [36] as used in QMR [18].

⁴Numerically, $p_k = A\bar{v}_k - \beta_k v_{k-1}$, $\alpha_k = v_k^* p_k$, $\beta_{k+1} v_{k+1} = p_k - \alpha_k v_k$ is slightly better [37].

In exact arithmetic, the columns of V_k are orthogonal and the process stops with $k = \ell$ and $\beta_{\ell+1} = 0$ for some $\ell \leq n$, and then $A\bar{V}_\ell = V_\ell T_\ell$. For derivation purposes we assume that this happens, though in practice it is rare unless V_k is reorthogonalized for each k . In any case, (2.2) holds to machine precision and the computed vectors satisfy $\|V_k\|_1 \simeq 1$ (even if $k \gg n$).

If instead we are given a skew symmetric A , the following is a Lanczos process [22, Algorithm 1]⁵ that transforms A to a series of expanding, skew symmetric tridiagonal matrices T_k and generates a set of orthogonal vectors in V_k in exact arithmetic:

$$p_k = Av_k, \quad -\beta_{k+1}v_{k+1} = p_k - \beta_k v_{k-1}. \quad (2.3)$$

Its associated matrix form is

$$AV_k = V_{k+1}\underline{T}_k, \quad \underline{T}_k \equiv \begin{bmatrix} 0 & \beta_2 & & & \\ -\beta_2 & 0 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & -\beta_k & 0 & \\ & & & -\beta_{k+1} & \end{bmatrix} \equiv \begin{bmatrix} T_k \\ -\beta_{k+1}e_k^T \end{bmatrix}. \quad (2.4)$$

If the skew symmetric process were to be forced on a skew Hermitian matrix, the resultant V_k would *not* be orthogonal. Instead, we multiply $Ax \approx b$ by i on both sides to yield a Hermitian problem since $(iA)^* = \bar{i}A^* = iA$. This simple transformation by a scalar multiplication⁶ preserves the condition for $\kappa(A) = \kappa(iA)$ and allows us to adapt the original Hermitian Lanczos process with $v_0 \equiv 0$, $\beta_1 v_1 = ib$, followed by

$$p_k = iAv_k, \quad \alpha_k = v_k^* p_k, \quad \beta_{k+1}v_{k+1} = p_k - \alpha_k v_k - \beta_k v_{k-1}. \quad (2.5)$$

Its matrix form is the same as (2.2) except that the first equation is $iAV_k = V_{k+1}\underline{T}_k$.

2.2. Properties of the Lanczos processes. The following properties of the Lanczos processes are notable:

1. If A and b real, then the Saunders process (2.1) reduces to the symmetric Lanczos process.
2. The complex and skew symmetric properties of A carry over to T_k by the Lanczos processes (2.1) and (2.3) respectively. From the skew Hermitian process (2.5), T_k is symmetric.
3. The skew-symmetric Lanczos process (2.3) is only two-term recurrent.
4. In (2.5), there are two ways to form p_k : $p_k = (iA)v_k$ or $p_k = A(iv_k)$. One may be cheaper than the other. If A is dense, iA takes $\mathcal{O}(n^2)$ scalar multiplications and storage. If A is sparse or structured as in the case of Toeplitz, iA just takes $\mathcal{O}(n)$ multiplications. In contrast, iv_k takes $n\ell$ multiplications, where ℓ is theoretically bounded by the number of distinct nonzero eigenvalues of A , but in practice ℓ could be an integer multiple of n .
5. While the skew Hermitian Lanczos process (2.5) is applicable to a skew symmetric problem, it involves complex arithmetic and is thus computationally more costly than the skew symmetric Lanczos process with a real b .
6. If A is changed to $A - \sigma I$ for some scalar shift σ , T_k becomes $T_k - \sigma I$ and V_k is unaltered, showing that singular systems are commonplace. Shifted

⁵Another Lanczos process for skew symmetric A uses a different measure to normalize β_{k+1} was developed in [53, 50].

⁶Multiplying by $-i$ works equally well but without loss of generality, we use i .

problems appear in inverse iteration or Rayleigh quotient iteration. The Lanczos frameworks easily and efficiently handles shifted problems.

7. Shifted skew symmetric matrices are not skew symmetric. This notion also applies to the case of shifted skew Hermitian matrices. Nevertheless they arise often in Toeplitz problems [3, 4].
8. For the skew Lanczos processes, the k th Krylov subspace generated by A and b is defined to be $\mathcal{K}_k(A, b) = \text{range}(V_k) = \text{span}\{b, Ab, \dots, A^{k-1}b\}$. For the Saunders process, we have a *modified* Krylov subspace [43] that we call the *Saunders subspace*, $\mathcal{K}_{k_1}(A\bar{A}, b) \oplus \mathcal{K}_{k_2}(A\bar{A}, A\bar{b})$, where $k_1 + k_2 = k$ and $0 \leq k_1 - k_2 \leq 1$.
9. T_k has full column rank k for all $k < \ell$ since $\beta_1, \dots, \beta_{k+1} > 0$.

THEOREM 2.1. T_ℓ is nonsingular if and only if $b \in \text{range}(A)$. Furthermore, $\text{rank}(T_\ell) = \ell - 1$ in the case $b \notin \text{range}(A)$.

Proof. We prove below for A complex symmetric. The proofs are similar for the skew symmetric and skew Hermitian cases.

We use $A\bar{V}_\ell = V_\ell T_\ell$ twice. First, if T_ℓ is nonsingular, we can solve $T_\ell y_\ell = \beta_1 e_1$ and then $A\bar{V}_\ell y_\ell = V_\ell T_\ell y_\ell = V_\ell \beta_1 e_1 = b$. Conversely, if $b \in \text{range}(A)$, then $\text{range}(\bar{V}_\ell) \subseteq \text{range}(\bar{A}) = \text{range}(A^*)$. Suppose T_ℓ is singular. Then there exists $z \neq 0$ such that $T_\ell z = 0$ and thus $V_\ell T_\ell z = A\bar{V}_\ell z = 0$. That is, $0 \neq \bar{V}_\ell z \in \text{null}(A)$. But this is impossible because $\bar{V}_\ell z \in \text{range}(A^*)$ and $\text{null}(A) \cap \text{range}(A^*) = \{0\}$. Thus T_ℓ must be nonsingular.

If $b \notin \text{range}(A)$, $T_\ell = \begin{bmatrix} T_{\ell-1} & \beta_\ell e_{\ell-1} \\ \alpha_\ell & \end{bmatrix}$ is singular. It follows that $\ell > \text{rank}(T_\ell) \geq \text{rank}(T_{\ell-1}) = \ell - 1$ since $\text{rank}(T_k) = k$ for all $k < \ell$. Therefore $\text{rank}(T_\ell) = \ell - 1$. \square

2.3. QLP decompositions for singular matrices. Here we generalize, from real to complex, the matrix decomposition *pivoted QLP* by Stewart in 1999 [49]⁷. It is equivalent to two consecutive QR factorizations with column interchanges, first on A , then on R^* :

$$Q_R A \Pi_R = \begin{bmatrix} R & S \\ 0 & 0 \end{bmatrix}, \quad Q_L \begin{bmatrix} R^* & 0 \\ S^* & 0 \end{bmatrix} \Pi_L = \begin{bmatrix} \hat{R} & 0 \\ 0 & 0 \end{bmatrix}, \quad (2.6)$$

giving *nonnegative* diagonal elements, where Π_R and Π_L are (real) permutations chosen to maximize the next diagonal element of R and \hat{R} at each stage. This gives

$$A = QLP, \quad Q = Q_R^* \Pi_L, \quad L = \begin{bmatrix} \hat{R}^* & 0 \\ 0 & 0 \end{bmatrix}, \quad P = Q_L \Pi_R^T,$$

with Q and P orthonormal. Stewart demonstrated that the diagonal elements of L (the *L-values*) give better singular-value estimates than those of R (the *R-values*), and the accuracy is particularly good for the extreme singular values σ_1 and σ_n :

$$R_{ii} \simeq \sigma_i, \quad L_{ii} \simeq \sigma_i, \quad \sigma_1 \geq \max_i L_{ii} \geq \max_i R_{ii}, \quad \min_i R_{ii} \geq \min_i L_{ii} \geq \sigma_n. \quad (2.7)$$

The first permutation Π_R in pivoted QLP is important. The main purpose of the second permutation Π_L is to ensure that the *L-values* present themselves in decreasing order, which is not always necessary. If $\Pi_R = \Pi_L = I$, it is simply called the *QLP decomposition*, which is applied to each T_k from the Lanczos processes (Section 2.1) in MINRES-QLP.

⁷QLP is a special case of the ULV decomposition, also by Stewart [48, 31].

2.4. Householder Reflectors. Givens rotations are often used to selectively annihilate matrix elements. Householder reflectors [51] of the following form may be considered the *Hermitian* counterpart of Givens rotations:

$$Q_{i,j} \equiv \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & c & \cdots & s & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & \bar{s} & \cdots & -c & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix},$$

where the subscripts indicate the positions of $c = \cos(\theta) \in \mathbb{R}$ and $s = \sin(\theta) \in \mathbb{C}$ for some angle θ . They are orthogonal and $Q_{i,j}^2 = I$ as would any reflector, meaning $Q_{i,j}$ is its own inverse. Thus $c^2 + |s|^2 = 1$. We often use the shorthand $Q_{i,j} = \begin{bmatrix} c & s \\ \bar{s} & -c \end{bmatrix}$.

In the next few sections we extend MINRES and MINRES-QLP to solving complex symmetric problems (1.1). Thus we tag the algorithms with “CS-”. The discussion and results can be easily adapted to the skew symmetric and skew Hermitian cases and so we do not go into details. In fact, the skew Hermitian problems can be solved by the implementations [10, 12] of MINRES and MINRES-QLP for Hermitian problems. For example, we can call the MATLAB solvers by $\mathbf{x} = \text{minres}(\mathbf{i} * \mathbf{A}, \mathbf{i} * \mathbf{b})$ and $\mathbf{x} = \text{minresqlp}(\mathbf{i} * \mathbf{A}, \mathbf{i} * \mathbf{b})$ achieving code reuse immediately.

3. CS-MINRES standalone. CS-MINRES is a natural way of using the complex symmetric Lanczos process (2.1) to solve (1.1). For $k < \ell$, if $x_k = \bar{V}_k y_k$ for some vector y_k , the associated residual is

$$r_k \equiv b - Ax_k = b - A\bar{V}_k y_k = \beta_1 v_1 - V_{k+1} \underline{T}_k y_k = V_{k+1}(\beta_1 e_1 - \underline{T}_k y_k). \quad (3.1)$$

To make r_k small, it is clear that $\beta_1 e_1 - \underline{T}_k y_k$ should be small. At this iteration k , CS-MINRES minimizes the residual subject to $x_k \in \text{range}(\bar{V}_k)$ by choosing

$$y_k = \arg \min_{y \in \mathbb{C}^k} \|\underline{T}_k y - \beta_1 e_1\|. \quad (3.2)$$

By Theorem 2.1, \underline{T}_k has full column rank and the above is a nonsingular problem.

3.1. QR factorization of \underline{T}_k . We apply an expanding QR factorization to the subproblem (3.2) by $Q_0 \equiv 1$ and

$$Q_{k,k+1} \equiv \begin{bmatrix} c_k & s_k \\ \bar{s}_k & -c_k \end{bmatrix}, \quad Q_k \equiv Q_{k,k+1} \begin{bmatrix} Q_{k-1} & \\ & 1 \end{bmatrix}, \quad Q_k [\underline{T}_k \quad \beta_1 e_1] = \begin{bmatrix} R_k & t_k \\ 0 & \phi_k \end{bmatrix}, \quad (3.3)$$

where c_k and s_k form the Householder reflector $Q_{k,k+1}$ that annihilates β_{k+1} in \underline{T}_k to give upper-tridiagonal R_k , with R_k and t_k being unaltered in later iterations. We

can express the last expression in (3.3) in terms of its elements for further analysis:

$$\begin{bmatrix} R_k \\ 0 \end{bmatrix} \equiv \begin{bmatrix} \gamma_1 & \delta_2 & \epsilon_3 & & \\ & \gamma_2^{(2)} & \delta_3^{(2)} & \ddots & \\ & & \ddots & \ddots & \epsilon_k \\ & & & \ddots & \delta_k^{(2)} \\ & & & & \gamma_k^{(2)} \\ & & & & 0 \end{bmatrix}, \quad \begin{bmatrix} t_k \\ \phi_k \end{bmatrix} \equiv \begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_k \\ \phi_k \end{bmatrix} = \beta_1 \begin{bmatrix} c_1 \\ \bar{s}_1 c_2 \\ \vdots \\ \bar{s}_1 \cdots \bar{s}_{k-1} c_k \\ \bar{s}_1 \cdots \bar{s}_{k-1} \bar{s}_k \end{bmatrix} \quad (3.4)$$

(where the superscripts are defined in Section 1.1). With $\phi_1 \equiv \beta_1 > 0$, the full action of $Q_{k,k+1}$ in (3.3), including its effect on later columns of T_i , $k < i \leq \ell$, is described by

$$\begin{bmatrix} c_k & s_k \\ \bar{s}_k & -c_k \end{bmatrix} \begin{bmatrix} \gamma_k & \delta_{k+1} & 0 \\ \beta_{k+1} & \alpha_{k+1} & \beta_{k+2} \end{bmatrix} \begin{bmatrix} \phi_{k-1} \\ 0 \end{bmatrix} = \begin{bmatrix} \gamma_k^{(2)} & \delta_{k+1}^{(2)} & \epsilon_{k+2} \\ 0 & \gamma_{k+1} & \delta_{k+2} \end{bmatrix} \begin{bmatrix} \tau_k \\ \phi_k \end{bmatrix}. \quad (3.5)$$

Thus for each $j \leq k < \ell$ we have $s_j \gamma_j^{(2)} = \beta_{j+1} > 0$, giving $\gamma_1, \gamma_j^{(2)} \neq 0$, and therefore each R_j is nonsingular. Also, $\tau_k = \phi_{k-1} c_k$ and $\phi_k = \phi_{k-1} \bar{s}_k \neq 0$. Hence from (3.1)–(3.3), we yield the following short recurrence relation for the residual norm:

$$\|r_k\| = \|T_k y_k - \beta_1 e_1\| = |\phi_k| \Rightarrow \|r_k\| = \|r_{k-1}\| |\bar{s}_k| = \|r_{k-1}\| |s_k|, \quad (3.6)$$

which is monotonically decreasing and tending to zero if $Ax = b$ is compatible.

3.2. Solving the subproblem. When $k < \ell$, a solution of (3.2) satisfies $R_k y_k = t_k$. Instead of solving for y_k , CS-MINRES solves $R_k^T D_k^T = V_k^*$ by forward substitution, obtaining the last column d_k of D_k at iteration k . This basis generation process can be summarized as

$$\begin{cases} d_1 = \bar{v}_1 / \gamma_1, & d_2 = (\bar{v}_2 - \delta_2 d_1) / \gamma_2^{(2)}, \\ d_k = (\bar{v}_k - \delta_k^{(2)} d_{k-1} - \epsilon_k d_{k-2}) / \gamma_k^{(2)}. \end{cases} \quad (3.7)$$

At the same time, CS-MINRES updates x_k via $x_0 \equiv 0$ and

$$x_k = \bar{V}_k y_k = D_k R_k y_k = D_k t_k = x_{k-1} + \tau_k d_k. \quad (3.8)$$

3.3. Termination. When $k = \ell$, we can form T_ℓ but nothing else expands. In place of (3.1) and (3.3) we have $r_\ell = V_\ell(\beta_1 e_1 - T_\ell y_\ell)$ and $Q_{\ell-1}[T_\ell \quad \beta_1 e_1] = [R_\ell \quad t_\ell]$. It is natural to solve for y_ℓ in the subproblem

$$\min_{y_\ell \in \mathbb{C}^\ell} \|T_\ell y_\ell - \beta_1 e_1\| \equiv \min_{y_\ell \in \mathbb{C}^\ell} \|R_\ell y_\ell - t_\ell\|. \quad (3.9)$$

There are two cases to consider:

1. If T_ℓ is nonsingular, $R_\ell y_\ell = t_\ell$ has a unique solution. Since $A \bar{V}_\ell y_\ell = V_\ell T_\ell y_\ell = b$, the problem $Ax = b$ is compatible and solved by $x_\ell = \bar{V}_\ell y_\ell$ with residual $r_\ell = 0$. Theorem 3.1 below proves that $x_\ell = x^\dagger$, assuring us that CS-MINRES is a useful solver for compatible linear systems even if A is singular.
2. If T_ℓ is singular, A and R_ℓ are singular ($R_{\ell\ell} = 0$) and both $Ax = b$ and $R_\ell y_\ell = t_\ell$ are incompatible. The optimal residual vector is unique, but infinitely many solutions give that residual. CS-MINRES sets the last element of y_ℓ to be zero. The final point and residual stay as $x_{\ell-1}$ and $r_{\ell-1}$ with $\|r_{\ell-1}\| = |\phi_{\ell-1}| = \beta_1 |s_1| \cdots |s_{\ell-1}| > 0$. Theorem 3.2 below proves that $x_{\ell-1}$ is a LS solution of $Ax \approx b$ (but not necessarily x^\dagger).

THEOREM 3.1. *If $b \in \text{range}(A)$, the final CS-MINRES point $x_\ell = x^\dagger$ and $r_\ell = 0$.*

Proof. If $b \in \text{range}(A)$, the Lanczos process gives $A\bar{V}_\ell = V_\ell T_\ell$ with nonsingular T_ℓ , and CS-MINRES terminates with $Ax_\ell = b$ and $x_\ell = \bar{V}_\ell y_\ell = A^*q = \bar{A}q$, where $q = V_\ell \bar{T}_\ell^{-1} y_\ell$. If some other point \hat{x} satisfies $A\hat{x} = b$, let $p = \hat{x} - x_\ell$. We have $Ap = 0$ and $x_\ell^* p = q^* Ap = 0$. Hence $\|\hat{x}\|^2 = \|x_\ell + p\|^2 = \|x_\ell\|^2 + 2x_\ell^* p + \|p\|^2 \geq \|x_\ell\|^2$. By (3.6), $r_\ell = 0$. \square

THEOREM 3.2. *If $b \notin \text{range}(A)$, then $\|Ar_{\ell-1}\| = 0$ and the CS-MINRES $x_{\ell-1}$ is an LS solution.*

Proof. Since $b \notin \text{range}(A)$, T_ℓ is singular and $R_{\ell\ell} = \gamma_\ell = 0$. By Lemma B.2, $A^*(Ax_{\ell-1} - b) = -\bar{A}r_{\ell-1} = -\|r_{\ell-1}\|\gamma_\ell v_\ell = 0$. Thus $x_{\ell-1}$ is an LS solution. \square

4. CS-MINRES-QLP standalone. In this section we develop CS-MINRES-QLP for solving ill-conditioned or singular symmetric systems. The Lanczos framework is the same as in CS-MINRES and QR factorization is applied to \underline{T}_k in subproblem (3.2) for all $k < \ell$; see Section 3.1. By Theorem 2.1 and Property 9 in Section 2.2, $\text{rank}(\underline{T}_k) = k$ for all $k < \ell$ and $\text{rank}(T_\ell) \geq \ell - 1$. CS-MINRES-QLP handles T_ℓ in (3.9) with extra care to *constructively* reveal $\text{rank}(T_\ell)$ via a QLP decomposition, so it can compute the minimum-length solution of the following subproblem instead of (3.9):

$$\min \|y_\ell\|_2 \quad \text{s.t.} \quad y_\ell \in \arg \min_{y_\ell \in \mathbb{C}^\ell} \|T_\ell y_\ell - \beta_1 e_1\|. \quad (4.1)$$

Thus CS-MINRES-QLP also applies the QLP decomposition on \underline{T}_k in (3.2) for all $k < \ell$.

4.1. QLP factorization of \underline{T}_k . In CS-MINRES-QLP, the QR factorization (3.3) of \underline{T}_k is followed by an LQ factorization of R_k :

$$Q_k \underline{T}_k = \begin{bmatrix} R_k \\ 0 \end{bmatrix}, \quad R_k P_k = L_k, \quad \text{so that} \quad Q_k \underline{T}_k P_k = \begin{bmatrix} L_k \\ 0 \end{bmatrix}, \quad (4.2)$$

where Q_k and P_k are orthogonal, R_k is upper tridiagonal, and L_k is lower tridiagonal. When $k < \ell$, both R_k and L_k are nonsingular. The QLP decomposition of each \underline{T}_k are performed without permutations, and the left and right reflectors interleaved [49] in order to ensure inexpensive updating of the factors as k increases. The desired rank-revealing properties (2.7) are retained in the last iteration when $k = \ell$.

We elaborate on interleaved QLP here. As in CS-MINRES, Q_k in (4.2) is a product of Householder reflectors, see (3.3) and (3.5), while P_k involves a product of *pairs* of Householder reflectors:

$$Q_k = Q_{k,k+1} \cdots Q_{3,4} \quad Q_{2,3} \quad Q_{1,2}, \quad P_k = P_{1,2} \quad P_{1,3} P_{2,3} \cdots P_{k-2,k} P_{k-1,k}.$$

For CS-MINRES-QLP to be efficient, in the k th iteration ($k \geq 3$) the application of the left reflector $Q_{k,k+1}$ is followed immediately by the right reflectors $P_{k-2,k}, P_{k-1,k}$, so that only the last 3×3 bottom right submatrix of \underline{T}_k is changed. These ideas can be understood more easily from the following compact form, which represents the actions of right reflectors on R_k obtained from (3.5):

$$\begin{aligned} & \begin{bmatrix} \gamma_{k-2}^{(5)} & & \epsilon_k \\ \vartheta_{k-1}^{(4)} & \gamma_{k-1}^{(4)} & \delta_k^{(2)} \\ & & \gamma_k^{(2)} \end{bmatrix} \begin{bmatrix} c_{k2} & s_{k2} \\ & 1 \\ \bar{s}_{k2} & -c_{k2} \end{bmatrix} \begin{bmatrix} 1 & & \\ & c_{k3} & s_{k3} \\ & \bar{s}_{k3} & -c_{k3} \end{bmatrix} \\ &= \begin{bmatrix} \gamma_{k-2}^{(6)} & & \\ \vartheta_{k-1}^{(2)} & \gamma_{k-1}^{(4)} & \delta_k^{(3)} \\ \eta_k & & \gamma_k^{(3)} \end{bmatrix} \begin{bmatrix} 1 & & \\ & c_{k3} & s_{k3} \\ & \bar{s}_{k3} & -c_{k3} \end{bmatrix} = \begin{bmatrix} \gamma_{k-2}^{(6)} & & \\ \vartheta_{k-1}^{(2)} & \gamma_{k-1}^{(5)} & \\ \eta_k & \vartheta_k & \gamma_k^{(4)} \end{bmatrix}. \quad (4.3) \end{aligned}$$

4.2. Solving the subproblem. With $y_k = P_k u_k$, subproblem (3.2) after QLP factorization of \underline{T}_k becomes

$$u_k = \arg \min_{u \in \mathbb{C}^k} \left\| \begin{bmatrix} L_k \\ 0 \end{bmatrix} u - \begin{bmatrix} t_k \\ \phi_k \end{bmatrix} \right\|, \quad (4.4)$$

where t_k and ϕ_k are as in (3.3). At the *start* of iteration k , the first $k-3$ elements of u_k , denoted by μ_j for $j \leq k-3$, are known from previous iterations. We need to solve only the last three components of u_k from the bottom three equations of $L_k u_k = t_k$:

$$\begin{bmatrix} \gamma_{k-2}^{(6)} & & & \\ \vartheta_{k-1}^{(2)} & \gamma_{k-1}^{(5)} & & \\ \eta_k & \vartheta_k & \gamma_k^{(4)} & \end{bmatrix} \begin{bmatrix} \mu_{k-2}^{(3)} \\ \mu_{k-1}^{(2)} \\ \mu_k \end{bmatrix} = \begin{bmatrix} \tau_{k-2}^{(2)} \\ \tau_{k-1}^{(2)} \\ \tau_k \end{bmatrix} \equiv \begin{bmatrix} \tau_{k-2} - \eta_{k-2} \mu_{k-4}^{(4)} - \vartheta_{k-2} \mu_{k-3}^{(3)} \\ \tau_{k-1} - \eta_{k-1} \mu_{k-3}^{(3)} \\ \tau_k \end{bmatrix}. \quad (4.5)$$

When $k < \ell$, \underline{T}_k has full column rank, and so do L_k and the above 3×3 triangular matrix. CS-MINRES-QLP obtains the same solution as CS-MINRES, but by a different process (and with different rounding errors). The CS-MINRES-QLP estimate of x is $x_k = \bar{V}_k y_k = \bar{V}_k P_k u_k = W_k u_k$, with theoretically orthonormal $W_k \equiv \bar{V}_k P_k$, where

$$\begin{aligned} W_k &= [\bar{V}_{k-1} P_{k-1} \quad \bar{v}_k] P_{k-2,k} P_{k-1,k} \\ &= \begin{bmatrix} W_{k-3}^{(4)} & w_{k-2}^{(3)} & w_{k-1}^{(2)} & \bar{v}_k \end{bmatrix} P_{k-2,k} P_{k-1,k} \\ &\equiv \begin{bmatrix} W_{k-3}^{(4)} & w_{k-2}^{(4)} & w_{k-1}^{(3)} & w_k^{(2)} \end{bmatrix}. \end{aligned} \quad (4.6)$$

Lastly, we update x_{k-2} and compute x_k by short-recurrence orthogonal steps (using only the last three columns of W_k):

$$x_{k-2}^{(2)} = x_{k-3}^{(2)} + w_{k-2}^{(4)} \mu_{k-2}^{(3)}, \text{ where } x_{k-3}^{(2)} \equiv W_{k-3}^{(4)} u_{k-3}^{(3)}, \quad (4.7)$$

$$x_k = x_{k-2}^{(2)} + w_{k-1}^{(3)} \mu_{k-1}^{(2)} + w_k^{(2)} \mu_k. \quad (4.8)$$

4.3. Termination. When $k = \ell$ and $y_\ell = P_\ell u_\ell$, the final subproblem (4.1) becomes

$$\min \|u_\ell\|_2 \quad \text{s.t.} \quad u_\ell \in \arg \min_{u_\ell \in \mathbb{C}^\ell} \|L_\ell u_\ell - t_\ell\|. \quad (4.9)$$

$Q_{\ell,\ell+1}$ is neither formed nor applied (see (3.3) and (3.5)), and the QR factorization stops. To obtain the minimum-length solution, we still need to apply $P_{\ell-2,\ell} P_{\ell-1,\ell}$ on the right of R_ℓ and \bar{V}_ℓ in (4.3) and (4.6) respectively. If $b \in \text{range}(A)$, then L_ℓ is nonsingular and the process in the previous subsection applies. If $b \notin \text{range}(A)$, the last row and column of L_ℓ are zero, i.e., $L_\ell = \begin{bmatrix} L_{\ell-1} & 0 \\ 0 & 0 \end{bmatrix}$ (see (4.2)), and we need to define $u_\ell \equiv \begin{bmatrix} u_{\ell-1} \\ 0 \end{bmatrix}$ and solve only the last two equations of $L_{\ell-1} u_{\ell-1} = t_{\ell-1}$:

$$\begin{bmatrix} \gamma_{\ell-2}^{(6)} & \\ \vartheta_{\ell-1}^{(2)} & \gamma_{\ell-1}^{(5)} \end{bmatrix} \begin{bmatrix} \mu_{\ell-2}^{(3)} \\ \mu_{\ell-1}^{(2)} \end{bmatrix} = \begin{bmatrix} \tau_{\ell-2}^{(2)} \\ \tau_{\ell-1}^{(2)} \end{bmatrix}. \quad (4.10)$$

The recurrence relation (4.8) simplifies to $x_\ell = x_{\ell-2}^{(2)} + w_{\ell-1}^{(3)} \mu_{\ell-1}^{(2)}$. The following theorem proves that CS-MINRES-QLP yields x^\dagger in this last iteration.

THEOREM 4.1. *In CS-MINRES-QLP, $x_\ell = x^\dagger$.*

Proof. When $b \in \text{range}(A)$, the proof is the same as that for Theorem 3.1.

When $b \notin \text{range}(A)$, for all $u = [u_{\ell-1}^T \ \mu_\ell]^T \in \mathbb{C}^\ell$ that solves (4.4), CS-MINRES-QLP returns the minimum-length LS solution $u_\ell = [u_{\ell-1}^T \ 0]^T$ by the construction in (4.10). For any $x \in \text{range}(W_\ell) = \text{range}(\bar{V}_\ell) \subseteq \text{range}(\bar{A}) = \text{range}(A^*)$ by (4.6) and $A\bar{V}_\ell = V_\ell T_\ell$,

$$\begin{aligned} \|Ax - b\| &= \|AW_\ell u - b\| = \|A\bar{V}_\ell P_\ell u - b\| = \|V_\ell T_\ell P_\ell u - \beta_1 V_\ell e_1\| = \|T_\ell P_\ell u - \beta_1 e_1\| \\ &= \left\| Q_{\ell-1} T_\ell P_\ell u - \begin{bmatrix} t_{\ell-1} \\ \phi_\ell \end{bmatrix} \right\| = \left\| \begin{bmatrix} L_{\ell-1} & 0 \\ 0 & 0 \end{bmatrix} u - \begin{bmatrix} t_{\ell-1} \\ \phi_\ell \end{bmatrix} \right\|. \end{aligned}$$

Since $L_{\ell-1}$ is nonsingular, $|\phi_\ell| = \min \|Ax - b\|$ can be achieved by $x_\ell = W_\ell u_\ell = W_{\ell-1} u_{\ell-1}$ and $\|x_\ell\| = \|W_{\ell-1} u_{\ell-1}\| = \|u_{\ell-1}\|$. Thus x_ℓ is the minimum-length LS solution of $\|Ax - b\|$, i.e., $x_\ell = \arg \min \{\|x\| \mid A^* Ax = A^* b, x \in \text{range}(A^*)\}$. Likewise $y_\ell = P_\ell u_\ell$ is the minimum-length LS solution of $\|T_\ell y - \beta_1 e_1\|$ and so $y_\ell \in \text{range}(T_\ell^*)$, i.e. $y_\ell = T_\ell^* z = \bar{T}_\ell z$ for some z . Thus $x_\ell = \bar{V}_\ell y_\ell = \bar{V}_\ell \bar{T}_\ell z = \bar{A} V_\ell z = A^* V_\ell z \in \text{range}(A^*)$. We know that $x^\dagger = \arg \min \{\|x\| \mid A^* Ax = A^* b, x \in \mathbb{C}^n\}$ is unique and $x^\dagger \in \text{range}(A^*)$. Since $x_\ell \in \text{range}(A^*)$, we must have $x_\ell = x^\dagger$. \square

5. Transferring CS-MINRES to CS-MINRES-QLP. CS-MINRES and CS-MINRES-QLP behave very similarly on well-conditioned systems. However, compared to CS-MINRES, CS-MINRES-QLP requires one more vector of storage, and each iteration needs 4 more axpy ($y \leftarrow \alpha x + y$) and 3 more vector scaling ($x \leftarrow \alpha x$). It would be a desirable feature to invoke CS-MINRES-QLP from CS-MINRES only if A is ill-conditioned or singular. The key idea is to transfer CS-MINRES to CS-MINRES-QLP at an iteration $k < \ell$ when \underline{T}_k has full column rank and is still well-conditioned. At such an iteration, the CS-MINRES point x_k^M and CS-MINRES-QLP point x_k are the same, so from (3.8), (4.8), and (4.4): $x_k^M = x_k \iff D_k t_k = W_k L_k^{-1} t_k$. From (3.7), (4.2), and (4.6),

$$D_k L_k = (\bar{V}_k R_k^{-1})(R_k P_k) = \bar{V}_k P_k = W_k. \quad (5.1)$$

The vertical arrow in Figure 5.1 represents this process. In particular, we can transfer only the last three CS-MINRES basis vectors in D_k to the last three CS-MINRES-QLP basis vectors in W_k :

$$\begin{bmatrix} w_{k-2} & w_{k-1} & w_k \end{bmatrix} = \begin{bmatrix} d_{k-2} & d_{k-1} & d_k \end{bmatrix} \begin{bmatrix} \gamma_{k-2}^{(6)} \\ \vartheta_{k-1}^{(2)} & \gamma_{k-1}^{(5)} \\ \eta_k & \vartheta_k & \gamma_k^{(4)} \end{bmatrix}. \quad (5.2)$$

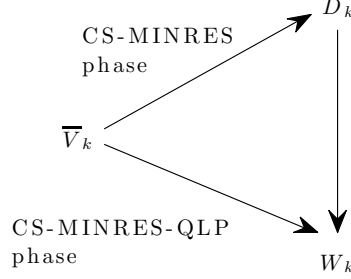
Furthermore, we need to generate the CS-MINRES-QLP point $x_{k-3}^{(2)}$ in (4.7) from the CS-MINRES point x_{k-1}^M by rearranging (4.8):

$$x_{k-3}^{(2)} = x_{k-1}^M - w_{k-2}^{(3)} \mu_{k-2}^{(2)} - w_{k-1}^{(2)} \mu_{k-1}. \quad (5.3)$$

Then the CS-MINRES-QLP points $x_{k-2}^{(2)}$ and x_k can be computed using (4.7) and (4.8).

It is clear from (5.1) and (5.2) that we still need to do the right transformation $R_k P_k = L_k$ in the CS-MINRES phase and keep the last 3×3 bottom right submatrix of L_k for each k so that we are ready to transfer to CS-MINRES-QLP when necessary. We then obtain a short recurrence for $\|x_k\|$ (see Section B.5) and for this computation we save flops relative to the standalone CS-MINRES algorithm, which computes $\|x_k\|$ directly in the NRBE condition associated with $\|r_k\|$ in Table B.1.

FIG. 5.1. *Changes of basis vectors within and between the two phases CS-MINRES and CS-MINRES-QLP; see equations (3.7), (4.6), and (5.2) for details.*



In the implementation of CS-MINRES-QLP, the iterates transfer from CS-MINRES to CS-MINRES-QLP when an estimate of the condition number of T_k (see (B.4)) exceeds an input parameter *trancond*. Thus, $\text{trancond} > 1/\varepsilon$ leads to CS-MINRES iterates throughout (that is, CS-MINRES standalone), while $\text{trancond} = 1$ generates CS-MINRES-QLP iterates from the start (that is, CS-MINRES-QLP standalone).

6. Preconditioned CS-MINRES and CS-MINRES-QLP. Well-constructed two-sided preconditioners can preserve problem symmetry and substantially reduce the number of iterations for nonsingular problems. For singular compatible problems, we can still solve the problems faster, but generally obtain only LS solutions that are not of minimum length—this is not an issue due to algorithms but the way two-sided preconditioning is set up for singular problems. For incompatible systems (which are necessarily singular), preconditioning alters the “least squares” norm. To avoid this difficulty we could work with larger equivalent systems that are compatible (see for example approaches in [8, Section 7.3]), or we could apply a right-hand-side preconditioner M preferably such that AM is complex symmetric so that our algorithms are directly applicable. For example, if M is nonsingular (complex) symmetric and AM is commutative, then $AMy \approx b$ is a complex symmetric problem with $y \equiv M^{-1}x$. This approach is efficient and straightforward. We devote the rest of this section to derive a two-sided preconditioning method.

We use a symmetric positive definite or a *nonsingular* complex symmetric preconditioner M . For such an M , it is known that the Cholesky factorization exists, i.e., $M = CC^T$ for some lower triangular matrix C , which is real if M is real, or complex if M is complex. We may employ commonly used construction techniques of preconditioners like diagonal preconditioning and incomplete Cholesky factorization if the nonzero entries of A are accessible. It may seem unnatural to use a symmetric positive-definite preconditioner for a complex symmetric problem. However, if available, its application may be less expensive than a complex symmetric preconditioner.

We denote the square root of M as $M^{-\frac{1}{2}}$. It is known that a complex symmetric root always exists for a nonsingular complex symmetric M even though it may not be unique; see [28, Theorems 7.1, 7.2, and 7.3] or [30, Section 6.4]. Preconditioned CS-MINRES (or CS-MINRES-QLP) applies itself to the equivalent system $\tilde{A}\tilde{x} = \tilde{b}$, where $\tilde{A} \equiv M^{-\frac{1}{2}}AM^{-\frac{1}{2}}$, $\tilde{b} \equiv M^{-\frac{1}{2}}b$, and $x = M^{-\frac{1}{2}}\tilde{x}$. Implicitly, we are solving an equivalent complex symmetric system $C^{-1}AC^{-T}y = C^{-1}b$, where $C^Tx = y$. In practice, we work with M itself (solving the linear system in (6.1)). For analysis, we

can assume $C = M^{\frac{1}{2}}$ for convenience. An effective preconditioner for CS-MINRES or CS-MINRES-QLP is one such that \tilde{A} has a more clustered eigenspectrum and become better conditioned, and it is inexpensive to solve linear systems that involve M .

6.1. Preconditioned Lanczos process. Let V_k denote the Lanczos vectors of the k th Krylov subspace generated by \tilde{A} and the conjugate of \tilde{b} . With $v_0 = 0$ and $\beta_1 v_1 = \tilde{b}$, for $k = 1, 2, \dots$ we define

$$z_k = \beta_k M^{\frac{1}{2}} v_k, \quad q_k = \beta_k M^{-\frac{1}{2}} \bar{v}_k, \quad \text{so that} \quad M \bar{q}_k = z_k. \quad (6.1)$$

Then $\beta_k = \|\beta_k v_k\| = \sqrt{q_k^T z_k}$, and the Lanczos iteration is

$$\begin{aligned} p_k &= \tilde{A} \bar{v}_k = M^{-\frac{1}{2}} A M^{-\frac{1}{2}} \bar{v}_k = M^{-\frac{1}{2}} A q_k / \beta_k, \\ \alpha_k &= v_k^* p_k = q_k^T A q_k / \beta_k^2, \\ \beta_{k+1} v_{k+1} &= M^{-\frac{1}{2}} A M^{-\frac{1}{2}} \bar{v}_k - \alpha_k v_k - \beta_k v_{k-1}. \end{aligned}$$

Multiplying the last equation by $M^{\frac{1}{2}}$ we get

$$\begin{aligned} z_{k+1} &= \beta_{k+1} M^{\frac{1}{2}} v_{k+1} = A M^{-\frac{1}{2}} \bar{v}_k - \alpha_k M^{\frac{1}{2}} v_k - \beta_k M^{\frac{1}{2}} v_{k-1} \\ &= \frac{1}{\beta_k} A q_k - \frac{\alpha_k}{\beta_k} z_k - \frac{\beta_k}{\beta_{k-1}} z_{k-1}. \end{aligned}$$

The last expression involving consecutive z_j 's replaces the three-term recurrence in v_j 's. In addition, we need to solve a linear system $M q_k = z_k$ (6.1) at each iteration.

6.2. Preconditioned CS-MINRES. From (3.8) and (3.7) we have the following recurrence for the k th column of $D_k = \bar{V}_k R_k^{-1}$ and \tilde{x}_k :

$$d_k = (\bar{v}_k - \delta_k^{(2)} d_{k-1} - \epsilon_k d_{k-2}) / \gamma_k^{(2)}, \quad \tilde{x}_k = \tilde{x}_{k-1} + \tau_k^{(2)} d_k.$$

Multiplying the above two equations by $M^{-\frac{1}{2}}$ on the left and defining $\tilde{d}_k = M^{-\frac{1}{2}} d_k$, we can update the solution of our original problem by

$$\tilde{d}_k = \left(\frac{1}{\beta_k} q_k - \delta_k^{(2)} \tilde{d}_{k-1} - \epsilon_k \tilde{d}_{k-2} \right) / \gamma_k^{(2)}, \quad x_k = M^{-\frac{1}{2}} \tilde{x}_k = x_{k-1} + \tau_k^{(2)} \tilde{d}_k.$$

6.3. Preconditioned CS-MINRES-QLP. A preconditioned CS-MINRES can be derived very similarly. The additional work is to apply right reflectors P_k to R_k , and the new subproblem bases are $\tilde{W}_k \equiv \bar{V}_k P_k$, with $\tilde{x}_k = \tilde{W}_k u_k$. Multiplying the new basis and solution estimate by $M^{-\frac{1}{2}}$ on the left, we obtain

$$\begin{aligned} \tilde{W}_k &\equiv M^{-\frac{1}{2}} \tilde{W}_k = M^{-\frac{1}{2}} \bar{V}_k P_k, \\ x_k &= M^{-\frac{1}{2}} \tilde{x}_k = M^{-\frac{1}{2}} \tilde{W}_k u_k = \tilde{W}_k u_k = x_{k-2}^{(2)} + \mu_{k-1}^{(2)} \tilde{w}_{k-1}^{(3)} + \mu_k \tilde{w}_k^{(2)}. \end{aligned}$$

7. Numerical experiments. In this section we present computational results based on MATLAB 7.12 implementation of CS-MINRES-QLP and SS-MINRES-QLP, which are made available to the public as open-source software and accords with the philosophy of reproducible computational research [13, 6]. The computations were performed in double precision on a Mac OS X machine with a 2.7GHz Intel Core i7 and 16GB RAM.

7.1. Complex symmetric problems. Although the SJSU Singular Matrix Database [17] currently contains only one complex symmetric matrix (named `dwg961a`) and only one skew symmetric matrix (`plsk1919`), it has a sizable set of singular symmetric matrices, which can be handled by the associated MATLAB toolbox `SJsingular` [16]. We constructed multiple singular complex symmetric systems of the form $H \equiv iA$, where A is symmetric and singular. It is clear that all the eigenvalues of H lie on the imaginary axis. For a compatible system, we simulated $b = Hz$, where $z_i \sim i.i.d. U(0, 1)$, i.e., z_i were independent and identically distributed random variables, whose values were drawn from the standard uniform distribution with support $[0, 1]$. For a LS problem, we generated a random b with $b_i \sim i.i.d. U(0, 1)$ and it is almost always true that b is *not* in the range of the test matrix. In CS-MINRES-QLP, we set the parameters `maxit` = $4n$, `tol` = ε , and `trancond` = 10^{-7} for the stopping conditions in Table B.1 and the transfer process from CS-MINRES (see Section 5).

We compare the computed results of CS-MINRES-QLP and MATLAB’s QMR to solutions computed directly by the truncated SVD (TSVD) of H utilizing MATLAB’s function `pinv`. For TSVD we have $x_t \equiv \sum_{\sigma_i > t\|H\|_\varepsilon} \frac{1}{\sigma_i} u_i u_i^* b$, with parameter $t > 0$. Often t is set to 1, and sometimes to a moderate number such as 10 or 100; it defines a cut-off point relative to the largest singular value of H . For example, if most singular values are of order 1 and the rest are of order $\|H\|_\varepsilon \approx 10^{-16}$, we expect TSVD to work better when the small singular values are excluded, while SVD (with $t = 0$) could return an exploding solution.

In Figure 7.1 we present the results of 50 consistent problems of the form $Hx = b$; it plots the relative error norm of approximate solutions computed by CS-MINRES-QLP and QMR with respect to TSVD solutions against a scalar multiple of $\kappa(H) = \kappa(A)$ —it is known that an upper bound of the perturbation error of a singular linear system involves the condition of the corresponding matrix [45, Theorem 5.1]. The diagonal dotted red line represents the best results we could expect from any numerical method with double precision. We can see that both CS-MINRES-QLP and QMR did very well on all problems except for one in each case. CS-MINRES-QLP performed slightly better as a few additional problems in QMR attained relative errors of less than 10^{-5} .

Our second test set involve complex symmetric matrices that have more widespread eigenspectrum than those in the first test set. Let $A = V\Lambda V^T$ be an eigenvalue decomposition of symmetric A with $|\lambda_1| \geq \dots \geq |\lambda_n|$. For $i = 1, \dots, n$, we define $d_i \equiv (2u_i - 1)|\lambda_1|$, where $u_i \sim i.i.d. U(0, 1)$ if $\lambda_i \neq 0$, or $d_i \equiv 0$ otherwise. Then the complex symmetric matrix $M \equiv VDV^T + iA$ has the same (numerical) rank as A and its eigenspectrum is bounded by a ball of radius approximately equal to $|\lambda_1|$ on the complex plane. In Figure 7.2 we summarize the results of solving 50 such complex symmetric linear systems. It is clear that CS-MINRES-QLP behaved as stably as it did with the first test set. However, QMR is obviously more sensitive to the nonlinear spectrum: two problems did not converge and about ten additional problems converged to their corresponding x^\dagger with no more than four digits of accuracy.

Our third test set consists of linear LS problems (1.1), in which $A \equiv H$ in the upper plot of Figure 7.3 and $A \equiv M$ in the lower plot. In the case of H , CS-MINRES-QLP did not converge for two instances but agreed with the TSVD solutions in five or more digits of accuracy for almost all other instances. In the case of M , CS-MINRES-QLP did not converge for five instances but agreed with the TSVD solutions in about five or more digits of accuracy for almost all other instances. Thus the algorithm is to some extent more sensitive to a nonlinear eigenspectrum in LS problems—this is also

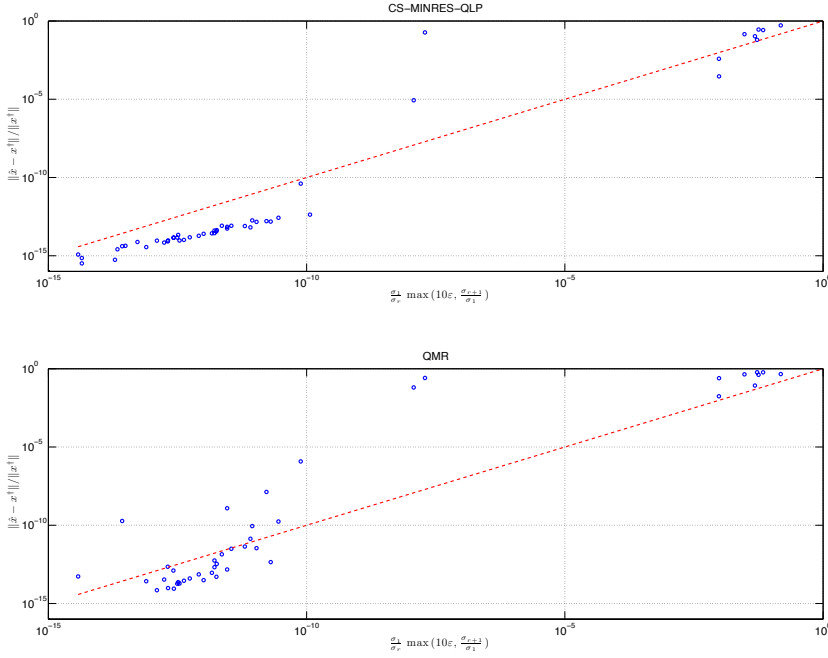


FIG. 7.1. 50 consistent singular complex symmetric systems. This figure is reproducible by C13Fig7.1.m.

expected by the perturbation result that an upper bound of the relative error norm in a LS problem involves the square of $\kappa(A)$ [45, Theorem 5.2]. We did not run QMR on these test cases since the algorithm was not designed for LS problems.

7.2. Skew symmetric problems. Our fourth test collection consists of 50 skew symmetric linear systems and 50 singular skew symmetric LS problems (1.1). The matrices are constructed by $S = \text{tril}(A) - \text{tril}(A)^T$, where tril extracts the lower triangular part of a matrix. In both cases—linear systems in the upper subplot of Figure 7.4 and LS problems in the lower subplot—SS-MINRES-QLP did not converge for six instances but agreed with the TSVD solutions for more than ten digits of accuracy for almost all other instances.

7.3. Skew Hermitian problems. We also have created a test collection that is constituted of 50 skew Hermitian linear systems and 50 skew Hermitian LS problems (1.1). Each of the skew Hermitian matrix is constructed by $T = S + iB$, where S is skew symmetric as defined in the last test set, and $B \equiv A - \text{diag}([a_{11}, \dots, a_{nn}])$, i.e., B is A with the diagonal elements set to zero and is thus symmetric. We solve the problems using the original MINRES-QLP for Hermitian problems by the transformation $(iT)x \approx ib$. In the case of linear systems in the upper subplot of Figure 7.5, SH-MINRES-QLP did not converge for six and five instances respectively. For the other instances SH-MINRES-QLP computed approximate solutions that matched the TSVD solutions for more than ten digits of accuracy for almost all other instances. As for LS problems in the lower subplot of Figure 7.5, only five instances did not converge.

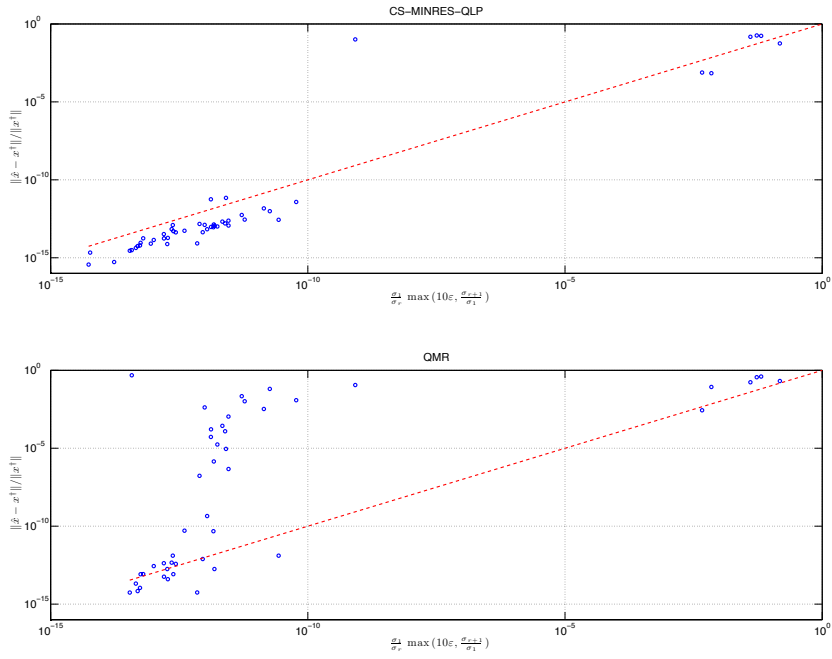


FIG. 7.2. 50 consistent singular complex symmetric systems. This figure is reproducible by C13Fig7.2.m.

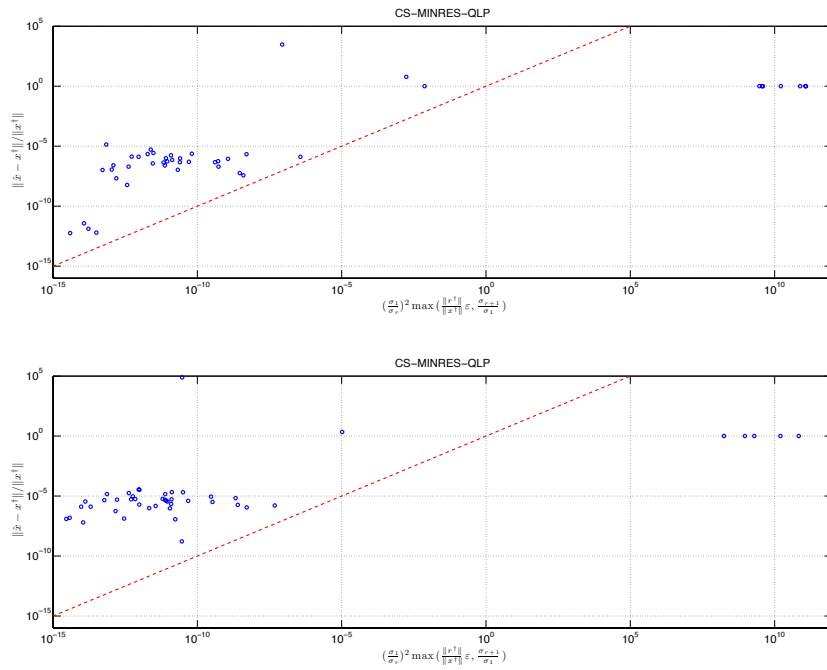


FIG. 7.3. 100 inconsistent singular complex symmetric systems. This figure is reproducible by C13Fig7.3.m.

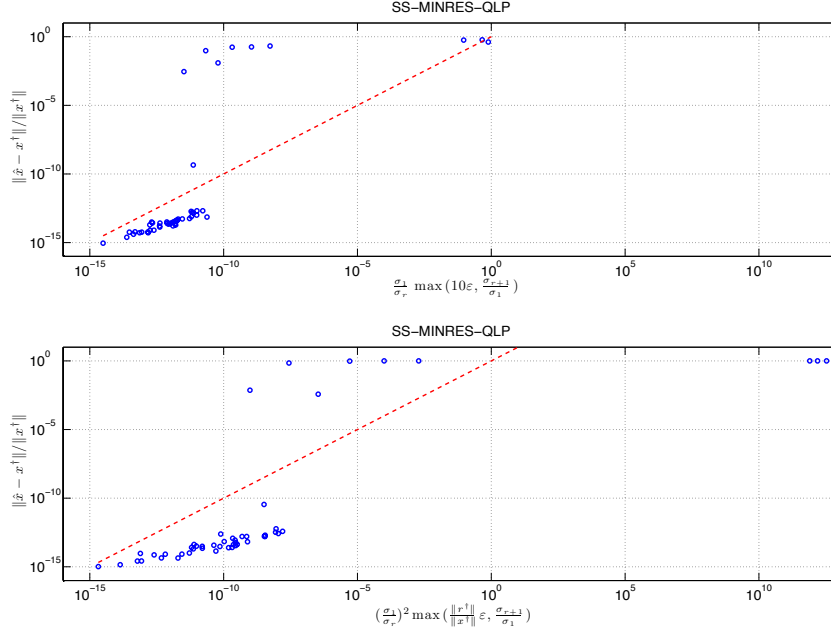


FIG. 7.4. 100 singular skew symmetric systems. Upper: 50 compatible linear systems. Lower: 50 LS problems. This figure is reproducible by C13Fig7.4.m.

8. Conclusions. CS-MINRES constructs its k th solution estimate from the short recursion $x_k = D_k t_k = x_{k-1} + \tau_k d_k$ (3.8), where n separate triangular systems $R_k^T D_k^T = V_k^*$ are solved to obtain the n elements of each direction d_1, \dots, d_k . (Only d_k is obtained during iteration k , but it has n elements.) In contrast, CS-MINRES-QLP constructs x_k using orthogonal steps: $x_k = W_k u_k = x_{k-2}^{(2)} + w_{k-1}^{(3)} \mu_{k-1}^{(2)} + w_k^{(2)} \mu_k$; see (4.7)–(4.8). Only one triangular system $L_k u_k = t_k$ (4.4) is involved for each k . Thus CS-MINRES-QLP is numerically more stable than CS-MINRES. In the mega form, the additional work and storage are moderate, and efficiency is retained by transferring from CS-MINRES to CS-MINRES-QLP only when the estimated condition number of A exceeds an input parameter value.

TSVD is known to use rank- k approximations to A to find approximate solutions to $\min \|Ax - b\|$ that serve as a form of *regularization*. It is fair to conclude from the results that like other Krylov methods CS-MINRES have built-in regularization features [26, 25, 35]. Since CS-MINRES-QLP monitors more carefully and constructively the rank of T_k , which could be k or $k-1$, we may say that regularization is a stronger feature in CS-MINRES-QLP, as we have shown in our numerical examples.

Like CS-MINRES and CS-MINRES-QLP, SS-MINRES and SS-MINRES-QLP are readily applicable to skew symmetric linear systems. We summarize and compare these methods in Appendix C.

8.1. Software and reproducible research. MATLAB 7.12 and Fortran 90/95 implementations of MINRES and MINRES-QLP for symmetric, Hermitian, skew symmetric, skew Hermitian, and complex symmetric linear systems with short-recurrence solution and norm estimates as well as efficient stopping conditions are available from the MINRES-QLP project website [10].

Following the philosophy of reproducible computational research as advocated

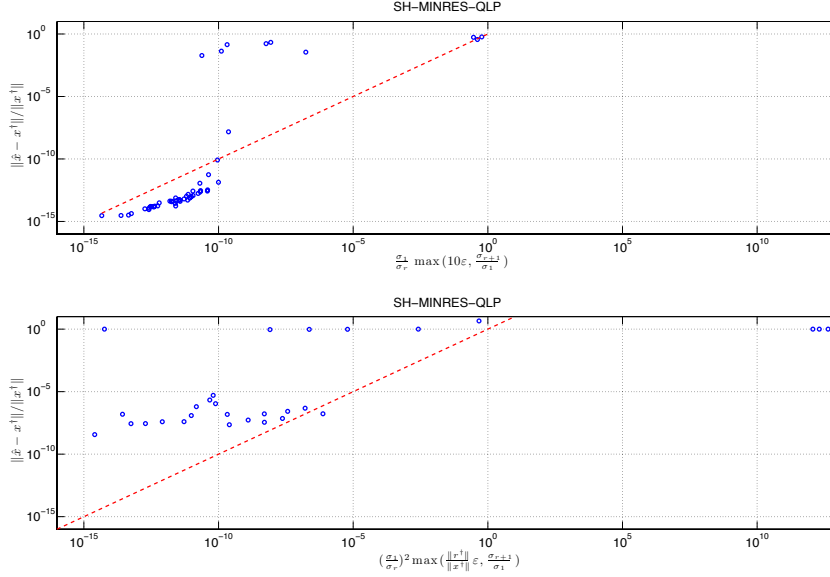


FIG. 7.5. 100 singular skew Hermitian systems. Upper: 50 compatible linear systems. Lower: 50 LS problems. This figure is reproducible by C13Fig7.5.m.

in [13, 6], for each figure and example in this paper we mention either the source or the specific MATLAB command. Our MATLAB scripts are available at [10].

Acknowledgements. We thank our anonymous reviewers for their insightful suggestions for improving this manuscript. We are grateful to the feedback and encouragement from Peter Benner, Jed Brown, Heike Fassbender, Gregory Fasshauer, Ian Foster, Roland Freund, Fred Hickernell, Ilse Ipsen, Sven Leyffer, Lek-Heng Lim, Sayan Mukherjee, Todd Munson, Nilima Nigam, Christopher Paige, Michael Saunders, Xiaobai Sun, Paul Tupper, Stefan Wild, and Jianlin Xia during the development of this work. We appreciate the opportunities for presenting our algorithms in the 2012 Copper Mountain Conference on Iterative Methods, the 2012 Structured Numerical Linear and Multilinear Algebra Problems conference, and the 2013 SIAM Conference on Computational Science and Engineering. In particular, we thank Gail Pieper and Daniel Szyld for their detailed comments, which resulted in a more polished exposition. We thank Heike Fassbender, Roland Freund, and Michael Saunders for helpful discussions on modified Krylov subspace methods during Lothar Reichel's 60th birthday conference.

Appendix A. Pseudocode of algorithms.**Algorithm 1: CS-Lanczos.**

input: $A, b, \sigma, \text{maxit}$

```

1  $v_0 = 0, \quad v_1 = b, \quad \beta_1 = \|b\|, \quad k = 0$ 
2 while  $k \leq \text{maxit}$  do
3   if  $\beta_{k+1} > 0$  then  $v_{k+1} \leftarrow v_{k+1}/\beta_{k+1}$  else STOP
4    $k \leftarrow k + 1$ 
5    $p_k = A\bar{v}_k - \sigma v_k$ 
6    $\alpha_k = v_k^* p_k$ 
7    $p_k \leftarrow p_k - \alpha_k v_k$ 
8    $v_{k+1} = p_k - \beta_k v_{k-1}$ 
9    $\beta_{k+1} = \|v_{k+1}\|$ 
output:  $V_\ell, T_\ell$ 

```

Algorithm 2: SS-Lanczos.

input: A, b, maxit

```

1  $v_0 = 0, \quad v_1 = b, \quad \beta_0 = 0, \quad \beta_1 = \|b\|, \quad k = 0$ 
2 while  $k \leq \text{maxit}$  do
3   if  $\beta_{k+1} > 0$  then  $v_{k+1} \leftarrow v_{k+1}/\beta_{k+1}$  else STOP
4    $k \leftarrow k + 1$ 
5    $p_k = Av_k - v_k$ 
6    $v_{k+1} = p_k - \beta_k v_{k-1}$ 
7    $\beta_{k+1} = \|v_{k+1}\|$ 
output:  $V_\ell, T_\ell$ 

```

Algorithm 3: SH-Lanczos.

input: A, b, maxit

```

1  $v_0 = 0, \quad v_1 = b, \quad \beta_1 = \|b\|, \quad k = 0$ 
2 while  $k \leq \text{maxit}$  do
3   if  $\beta_{k+1} > 0$  then  $v_{k+1} \leftarrow v_{k+1}/\beta_{k+1}$  else STOP
4    $k \leftarrow k + 1$ 
5    $p_k = iAv_k - iv_k$ 
6    $\alpha_k = v_k^* p_k$ 
7    $p_k \leftarrow p_k - \alpha_k v_k$ 
8    $v_{k+1} = p_k - \beta_k v_{k-1}$ 
9    $\beta_{k+1} = \|v_{k+1}\|$ 
output:  $V_\ell, T_\ell$ 

```

Appendix B. Stopping conditions and norm estimates.

This section derives several short-recurrence norm estimates in CS-MINRES and SH-MINRES. As before, we assume exact arithmetic throughout, so that V_k and Q_k are orthonormal. Table B.1 summarizes how these norm estimates are used to formulate six groups of concerted stopping conditions. The second NRBE test is specifically designed for LS problems, which have the properties $r \neq 0$ but $A^*r = 0$; it is inspired by Stewart [47] and stops the algorithm when $\|A^*r_k\|$ is small relative to its upper bound $\|A\|\|r_k\|$.

Algorithm 4: Algorithm SymOrtho.

input: a, b

- 1 **if** $|b| = 0$ **then** $c = 1, \quad s = 0, \quad r = a$
- 2 **else if** $|a| = 0$ **then**
- 3 $c = 0, \quad s = 1, \quad r = b$
- 4 **else if** $|b| \geq |a|$ **then**
- 5 $\tau = |a| / |b|, \quad c = 1 / \sqrt{1 + \tau^2}, \quad s = c\tau(b/a), \quad c \leftarrow c\tau, \quad r = b/\bar{s}$
- 6 **else if** $|a| > |b|$ **then**
- 7 $\tau = |b| / |a|, \quad c = 1 / \sqrt{1 + \tau^2}, \quad s = c(b/a), \quad r = a/c$

output: c, s, r

B.1. Residual and residual norm. First we derive recurrence relations for r_k and its norm $\|r_k\| = |\phi_k|$. The results are true for CS-MINRES and CS-MINRES-QLP.

LEMMA B.1. *Without loss of generality, let $x_0 = 0$. The following results follow.*

1. $r_0 = b$ and $\|r_0\| = \phi_0 = \beta_1$.
2. For $k = 1, \dots, \ell - 1$, then $\|r_k\| = |\phi_k| = |\phi_{k-1}|s_k| \geq |\phi_{k-1}| > 0$. Thus $\|r_k\|$ is monotonically decreasing.
3. At the last iteration ℓ ,
 - (a) If $\text{rank}(L_\ell) = \ell$, then $\|r_\ell\| = \phi_\ell = 0$.
 - (b) If $\text{rank}(L_\ell) = \ell - 1$, then $\|r_\ell\| = |\phi_{\ell-1}| > 0$.

Proof.

1. Obvious.
2. If $k < \ell$, From (3.1)–(3.8) with $R_k y_k = t_k$ we have

$$r_k = V_{k+1} Q_k^* \left(\begin{bmatrix} t_k \\ \phi_k \end{bmatrix} - \begin{bmatrix} R_k \\ 0 \end{bmatrix} y_k \right) = \phi_k V_{k+1} Q_k^* e_{k+1}. \quad (\text{B.1})$$

We have $\|r_k\| = |\phi_k| = |\phi_{k-1}|s_k| > 0$; see (3.4)–(3.6).

3. If T_ℓ is nonsingular, $r_\ell = 0$. Otherwise $Q_{\ell-1, \ell}$ has made the last row of R_ℓ zero, so the last row and column of L_ℓ are zero; see (4.10). Thus $r_\ell = r_{\ell-1} \neq 0$.

□

B.2. Norm of $A^* r_k$. For incompatible systems, r_k will never be zero. However, all LS solutions satisfy $A^* A x = A^* b$, so that $A^* r = 0$. We therefore need a stopping condition based on the size of $\|A^* r_k\| = \psi_k$. We present efficient recurrence relations for $\|A^* r_k\|$ in the following Lemma. We also show that $A^* r_k$ is orthogonal to $\mathcal{K}_k(A, \bar{b})$.

LEMMA B.2 ($A^* r_k$ and $\psi_k \equiv \|A^* r_k\|$ for CS-MINRES).

1. If $k < \ell$, then $\text{rank}(L_k) = k$, $\bar{A} r_k = \|r_k\|(\bar{\gamma}_{k+1} \bar{v}_{k+1} + \delta_{k+2} \bar{v}_{k+2})$ and $\psi_k = \|r_k\| \|\begin{bmatrix} \gamma_{k+1} & \delta_{k+2} \end{bmatrix}\|$, where $\delta_{k+2} = 0$ if $k = \ell - 1$.
2. At the last iteration ℓ ,
 - (a) If $\text{rank}(L_\ell) = \ell$, then $A r_\ell = 0$ and $\psi_\ell = 0$.
 - (b) If $\text{rank}(L_\ell) = \ell - 1$, then $A r_\ell = A r_{\ell-1} = 0$, and $\psi_\ell = \psi_{\ell-1} = 0$.

Algorithm 5: Preconditioned CS-MINRES-QLP to solve $(A - \sigma I)x \approx b$.

input: A, b, σ, M

1 $z_0 = 0, \quad z_1 = b, \quad \text{Solve } Mq_1 = z_1, \quad \beta_1 = \sqrt{b^T q_1} \quad [\text{Initialize}]$
2 $w_0 = w_{-1} = 0, \quad x_{-2} = x_{-1} = x_0 = 0$
3 $c_{0,1} = c_{0,2} = c_{0,3} = -1, \quad s_{0,1} = s_{0,2} = s_{0,3} = 0, \quad \phi_0 = \beta_1, \quad \tau_0 = \omega_0 = \chi_{-2} = \chi_{-1} = \chi_0 = 0$
4 $\delta_1 = \gamma_{-1} = \gamma_0 = \eta_{-1} = \eta_0 = \eta_1 = \vartheta_{-1} = \vartheta_0 = \vartheta_1 = \mu_{-1} = \mu_0 = 0, \quad \mathcal{A} = 0, \quad \kappa = 1$
5 $k = 0$

6 **while** no stopping condition is satisfied **do**

7 $k \leftarrow k + 1$

8 $p_k = Aq_k - \sigma q_k, \quad \alpha_k = \frac{1}{\beta_k^2} q_k^T p_k \quad [\text{Preconditioned Lanczos}]$
9 $z_{k+1} = \frac{1}{\beta_k} p_k - \frac{\alpha_k}{\beta_k} z_k - \frac{\beta_k}{\beta_{k-1}} z_{k-1}$
10 $\text{Solve } Mq_{k+1} = z_{k+1}, \quad \beta_{k+1} = \sqrt{q_{k+1}^T z_{k+1}}$
11 **if** $k = 1$ **then** $\rho_k = \|[\alpha_k \ \beta_{k+1}]\|$ **else** $\rho_k = \|[\beta_k \ \alpha_k \ \beta_{k+1}]\|$
12 $\delta_k^{(2)} = c_{k-1,1} \delta_k + s_{k-1,1} \alpha_k \quad [\text{Previous left reflection...}]$
13 $\gamma_k = s_{k-1,1} \delta_k - c_{k-1,1} \alpha_k \quad [\text{on middle two entries of } T_k e_k \dots]$
14 $\epsilon_{k+1} = s_{k-1,1} \beta_{k+1} \quad [\text{produces first two entries in } T_{k+1} e_{k+1}]$
15 $\delta_{k+1} = -c_{k-1,1} \beta_{k+1}$
16 $c_{k1}, s_{k1}, \gamma_k^{(2)} \leftarrow \text{SymOrtho}(\gamma_k, \beta_{k+1}) \quad [\text{Current left reflection}]$
17 $c_{k2}, s_{k2}, \gamma_{k-2}^{(6)} \leftarrow \text{SymOrtho}(\gamma_{k-2}^{(5)}, \epsilon_k) \quad [\text{First right reflection}]$
18 $\delta_k^{(3)} = s_{k2} \vartheta_{k-1} - c_{k2} \delta_k^{(2)}, \quad \gamma_k^{(3)} = -c_{k2} \gamma_k^{(2)}, \quad \eta_k = s_{k2} \gamma_k^{(2)}$
19 $\vartheta_{k-1}^{(2)} = c_{k2} \vartheta_{k-1} + s_{k2} \delta_k^{(2)}$
20 $c_{k3}, s_{k3}, \gamma_{k-1}^{(5)} \leftarrow \text{SymOrtho}(\gamma_{k-1}^{(4)}, \delta_k^{(3)}) \quad [\text{Second right reflection...}]$
21 $\vartheta_k = s_{k3} \gamma_k^{(3)}, \quad \gamma_k^{(4)} = -c_{k3} \gamma_k^{(3)} \quad [\text{to zero out } \delta_k^{(3)}]$
22 $\tau_k^{(2)} = c_{k1} \tau_k \quad [\text{Last element of } t_k]$
23 $\phi_k = \phi_{k-1} |s_{k1}|, \quad \psi_{k-1} = \phi_{k-1} \|[\gamma_k \ \delta_{k+1}]\| \quad [\text{Update } \|r_k\|, \|Ar_{k-1}\|]$
24 **if** $k = 1$ **then** $\gamma_{\min} = \gamma_1$ **else** $\gamma_{\min} \leftarrow \min\{\gamma_{\min}, \gamma_{k-2}^{(6)}, \gamma_{k-1}^{(5)}, |\gamma_k^{(4)}|\}$
25 $\mathcal{A}^{(k)} = \max\{\mathcal{A}^{(k-1)}, \rho_k, \gamma_{k-2}^{(6)}, \gamma_{k-1}^{(5)}, |\gamma_k^{(4)}|\} \quad [\text{Update } \|\mathcal{A}\|]$
26 $\omega_k = \|[\omega_{k-1} \ \tau_k^{(2)}]\|, \quad \kappa \leftarrow \mathcal{A}^{(k)} / \gamma_{\min} \quad [\text{Update } \|Ax_k\|, \kappa(A)]$
27 $w_k = -(c_{k2}/\beta_k) q_k + s_{k2} w_{k-2}^{(3)} \quad [\text{Update } w_{k-2}, w_{k-1}, w_k]$
28 $w_{k-2}^{(4)} = (s_{k2}/\beta_k) q_k + c_{k2} w_{k-2}^{(3)}$
29 **if** $k > 2$ **then** $w_k^{(2)} = s_{k3} w_{k-1}^{(2)} - c_{k3} w_k, \quad w_{k-1}^{(3)} = c_{k3} w_{k-1}^{(2)} + s_{k3} w_k$
30 **if** $k > 2$ **then** $\mu_{k-2}^{(3)} = (\tau_{k-2}^{(2)} - \eta_{k-2} \mu_{k-4}^{(4)} - \vartheta_{k-2} \mu_{k-3}^{(3)}) / \gamma_{k-2}^{(6)} \quad [\text{Update } \mu_{k-2}]$
31 **if** $k > 1$ **then** $\mu_{k-1}^{(2)} = (\tau_{k-1}^{(2)} - \eta_{k-1} \mu_{k-3}^{(3)} - \vartheta_{k-1} \mu_{k-2}^{(3)}) / \gamma_{k-1}^{(5)} \quad [\text{Update } \mu_{k-1}]$
32 **if** $\gamma_k^{(4)} \neq 0$ **then** $\mu_k = (\tau_k^{(2)} - \eta_k \mu_{k-2}^{(3)} - \vartheta_k \mu_{k-1}^{(2)}) / \gamma_k^{(4)}$ **else** $\mu_k = 0$ **[Compute } μ_k $]$**

33 $x_{k-2}^{(2)} = x_{k-3}^{(2)} + \mu_{k-2}^{(3)} w_{k-2}^{(3)} \quad [\text{Update } x_{k-2}]$
34 $x_k = x_{k-2}^{(2)} + \mu_{k-1}^{(2)} w_{k-1}^{(3)} + \mu_k w_k^{(2)} \quad [\text{Compute } x_k]$
35 $\chi_{k-2}^{(2)} = \|[\chi_{k-3}^{(2)} \ \mu_{k-2}^{(3)}]\| \quad [\text{Update } \|x_{k-2}\|]$
36 $\chi_k = \|[\chi_{k-2}^{(2)} \ \mu_{k-1}^{(2)} \ \mu_k]\| \quad [\text{Compute } \|x_k\|]$

37 $x = x_k, \quad \phi = \phi_k, \quad \psi = \phi_k \|[\gamma_{k+1} \ \delta_{k+2}]\|, \quad \chi = \chi_k, \quad \mathcal{A} = \mathcal{A}^{(k)}, \quad \omega = \omega_k$

output: $x, \phi, \psi, \chi, \mathcal{A}, \kappa, \omega$

38 $[c, s \leftarrow \text{SymOrtho}(a, b)]$ is a stable form for computing $r = \sqrt{a^2 + b^2}$, $c = \frac{a}{r}$, $s = \frac{b}{r}$

TABLE B.1

Stopping conditions in CS-MINRES and SH-MINRES. NRBE means normwise relative backward error, and tol , $maxit$, $maxcond$ and $maxnorm$ are input parameters. All norms and $\kappa(A)$ are estimated by CS-MINRES and SH-MINRES.

Lanczos	NRBE	Regularization attempts
$\beta_{k+1} \leq n\ A\ \varepsilon$	$\ r_k\ / (\ A\ \ x_k\ + \ b\) \leq \max(tol, \varepsilon)$	$\kappa(A) \geq \max(maxcond, \frac{1}{\varepsilon})$
$k = maxit$	$\ \bar{A}r_k\ / (\ A\ \ r_k\) \leq \max(tol, \varepsilon)$	$\ x_k\ \geq maxnorm$
CS-MINRES	Degenerate cases	Erroneous input
$ \gamma_k^{(4)} < \varepsilon$	$\beta_1 = 0 \Rightarrow x^\dagger = 0$ $\beta_2 = 0 \Rightarrow x^\dagger = \bar{b}/\alpha_1$	$y^*(Az) \neq \pm z^*(\bar{A}y)$ $\Rightarrow A \neq \pm A^T$

Proof. Case 2 follows directly from Lemma B.1. We prove the first case here. For $k < \ell$, R_k is nonsingular. From (3.1)–(3.8) with $R_k y_k = t_k$ we have

$$\bar{A}r_k = \phi_k \bar{V}_{k+2} \bar{T}_{k+1} Q_k^* e_{k+1}, \text{ by (B.1)}$$

$$Q_k \bar{T}_{k+1}^T = Q_k [T_{k+1} \quad \beta_{k+2} e_{k+1}] = Q_k \begin{bmatrix} T_k & \beta_{k+1} e_k & 0 \\ \beta_{k+1} e_k^T & \alpha_{k+1} & \beta_{k+2} \end{bmatrix},$$

$$e_{k+1}^T Q_k \bar{T}_{k+1}^T = [0 \quad \gamma_{k+1} \quad \delta_{k+2}],$$

by (3.5). We take $\delta_{k+2} = 0$ if $k = \ell - 1$, so

$$\bar{A}r_k = \tau_{k+1} \bar{V}_{k+2} [0 \quad \gamma_{k+1} \quad \delta_{k+2}]^* = \tau_{k+1} (\bar{\gamma}_{k+1} \bar{v}_{k+1} + \delta_{k+2} \bar{v}_{k+2}),$$

$$\psi_k^2 \equiv \|\bar{A}r_k\|^2 = \|r_k\|^2 ([\gamma_{k+1}]^2 + [\delta_{k+2}]^2).$$

Result follows. \square

Typically $\|\bar{A}r_k\|$ is not monotonic, while clearly $\|r_k\|$ is monotonically decreasing. In the singular system $A = U\Sigma U^T$, let $U = [U_1 \ U_2]$, where the singular vectors U_1 correspond to nonzero singular values. Then $P_A \equiv U_1 U_1^*$ and $P_A^\perp \equiv U_2 U_2^*$ are orthogonal projectors [51] onto the range and nullspace of A . For general linear LS problems, Chang et al. [5] characterize the dynamics of $\|r_k\|$ and $\|A^* r_k\|$ in three phases defined in terms of the ratios among $\|r_k\|$, $\|P_A r_k\|$ and $\|P_A^\perp r_k\|$, and propose two new stopping criteria for iterative solvers. The expositions in [1, 34] show that these estimates are cheaply computable in CGLS and LSQR [39, 40]. These results are likely applicable to CS-MINRES.

B.3. Matrix norms. From the Lanczos process, $\|A\| \geq \|V_{k+1}^* A \bar{V}_k\| = \|\underline{T}_k\|$. Define

$$\mathcal{A}^{(0)} \equiv 0, \quad \mathcal{A}^{(k)} \equiv \max_{j=1, \dots, k} \left\{ \|\underline{T}_j e_j\| \right\} = \max \left\{ \mathcal{A}^{(k-1)}, \|\underline{T}_k e_k\| \right\} \text{ for } k \geq 1. \quad (\text{B.2})$$

Then $\|A\| \geq \|\underline{T}_k\| \geq \mathcal{A}^{(k)}$. Clearly, $\mathcal{A}^{(k)}$ is monotonically increasing and is thus an improving estimate for $\|A\|$ as k increases. By the property of QLP decomposition in (2.7) and (4.3), we could easily extend (B.2) to include the largest diagonal of L_k :

$$\mathcal{A}^{(0)} \equiv 0, \quad \mathcal{A}^{(k)} \equiv \max \{ \mathcal{A}^{(k-1)}, \|\underline{T}_k e_k\|, \gamma_{k-2}^{(6)}, \gamma_{k-1}^{(5)}, |\gamma_k^{(4)}| \} \text{ for } k \geq 1, \quad (\text{B.3})$$

which uses quantities readily available from CS-MINRES and gives satisfactory, if not extremely accurate, estimates for the order of $\|A\|$.

B.4. Matrix condition numbers. We again apply the property of the QLP decomposition in (2.7) and (4.3) to estimate $\kappa(\underline{T}_k)$, which is a lower bound for $\kappa(A)$:

$$\begin{aligned} \gamma_{\min} &\leftarrow \min\{\gamma_1, \gamma_2^{(2)}\}, \quad \gamma_{\min} \leftarrow \min\{\gamma_{\min}, \gamma_{k-2}^{(6)}, \gamma_{k-1}^{(5)}, |\gamma_k^{(4)}|\} \text{ for } k \geq 3, \\ \kappa^{(0)} &\equiv 1, \quad \kappa^{(k)} \equiv \max\left\{\kappa^{(k-1)}, \frac{\mathcal{A}^{(k)}}{\gamma_{\min}}\right\} \text{ for } k \geq 1. \end{aligned} \quad (\text{B.4})$$

B.5. Solution norms. For CS-MINRES-QLP, we derive a recurrence relation for $\|x_k\|$ whose cost is as low as computing the norm of a 3- or 4- vector. This recurrence relation is not applicable to CS-MINRES standalone.

Since $\|x_k\| = \|\bar{V}_k P_k u_k\| = \|u_k\|$, we can estimate $\|x_k\|$ by computing $\chi_k \equiv \|u_k\|$. However, the last two elements of u_k change in u_{k+1} (and a new element μ_{k+1} is added). We therefore maintain χ_{k-2} by updating it and then using it according to

$$\chi_{k-2}^{(2)} = \|[\chi_{k-3}^{(2)} \quad \mu_{k-2}^{(3)}]\|, \quad \chi_k = \|[\chi_{k-2}^{(2)} \quad \mu_{k-1}^{(2)} \quad \mu_k]\| \quad \text{cf. (4.7) and (4.8)}.$$

Thus $\chi_{k-2}^{(2)}$ increases monotonically but we cannot guarantee that $\|x_k\|$ and its recurred estimate χ_k are increasing, and indeed they are not in some examples. But the trend for χ_k is generally increasing, and $\chi_k^{(2)}$ is theoretically a better estimate than χ_k for $\|x_k\|$. In LS problems, when $\gamma_k^{(4)}$ is small enough in magnitude, it also means $\|x_k\| = \|y_k\| = \|u_k\|$ is large—and when this quantity is larger than maxnorm , it usually means that we should do only a partial update of $x_k = x_{k-2}^{(2)} + w_{k-1}^{(3)} \mu_{k-1}^{(2)}$, which if still exceeds maxnorm in length, then we shall do no update, i.e., $x_k = x_{k-2}^{(2)}$.

B.6. Projection norms. In applications requiring nullvectors [7], Ax_k is useful. Other times, the projection of the right-hand side b onto $\mathcal{K}_k(A, \bar{b})$ is required [42]. For the recurrence relations of Ax_k and its norm, we have

$$\begin{aligned} Ax_k &= A\bar{V}_k y_k = V_{k+1} \underline{T}_k y_k = V_{k+1} Q_k^* \begin{bmatrix} R_k \\ 0 \end{bmatrix} y_k = V_{k+1} Q_k^* \begin{bmatrix} t_k \\ 0 \end{bmatrix}, \\ \omega_k^2 &\equiv \|Ax_k\|^2 = \|t_k\|^2 = \|t_{k-1}\|^2 + (\tau_k^{(2)})^2 = \omega_{k-1}^2 + (\tau_k^{(2)})^2 = \|[\omega_{k-1} \quad \tau_k^{(2)}]\|^2. \end{aligned}$$

Thus $\{\omega_k\}$ is monotonic.

Appendix C. Comparison of Lanczos-based solvers.

We compare our new solvers with CG, SYMMLQ, MINRES, and MINRES-QLP in Tables C.1–C.2 in terms of subproblem definitions, basis, solution estimates, flops, and memory. A careful implementation of SYMMLQ computes x_k in $\text{range}(V_{k+1})$; see [7, Section 2.2.2] for a proof. All solvers need storage for v_k , v_{k+1} , x_k , and a product $p_k = Av_k$ or $A\bar{v}_k$ each iteration. Some additional work-vectors are needed for each method (e.g., d_{k-1} and d_k for MINRES or CS-MINRES, giving 7 work-vectors in total). It is noteworthy that even for Hermitian and skew Hermitian problems $Ax = b$, the subproblems of CG, SYMMLQ, MINRES, and MINRES-QLP are real.

TABLE C.1

Subproblems defining x_k for CG, SYMMLQ, MINRES, MINRES-QLP, CS-MINRES, CS-MINRES-QLP, SH-MINRES, and SH-MINRES-QLP.

Method	Subproblem	Factorization	Estimate of x_k
cgLanczos [27, 38, 44]	$T_k y_k = \beta_1 e_1$	Cholesky: $T_k = L_k D_k L_k^T$	$x_k = V_k y_k$ $\in \text{range}(V_k)$
SYMMLQ [38, 7]	$y_{k+1} = \arg \min_{y \in \mathbb{R}^{k+1}} \ y\ $ s.t. $\underline{T}_k^T y = \beta_1 e_1$	LQ: $\underline{T}_k^T Q_k^T = [L_k \ 0]$	$x_k = V_{k+1} y_{k+1}$ $\in \text{range}(V_{k+1})$
MINRES [38],[7]-[11]	$y_k = \arg \min_{y \in \mathbb{R}^k} \ \underline{T}_k y - \beta_1 e_1\ $	QR: $Q_k \underline{T}_k = \begin{bmatrix} R_k \\ 0 \end{bmatrix}$	$x_k = V_k y_k$ $\in \text{range}(V_k)$
MINRES-QLP [7]-[11]	$y_k = \arg \min_{y \in \mathbb{R}^k} \ y\ $ s.t. $y \in \arg \min \ \underline{T}_k y - \beta_1 e_1\ $	QLP: $Q_k \underline{T}_k P_k = \begin{bmatrix} L_k \\ 0 \end{bmatrix}$	$x_k = V_k y_k$ $\in \text{range}(V_k)$
CS-MINRES	$y_k = \arg \min_{y \in \mathbb{C}^k} \ \underline{T}_k y - \beta_1 e_1\ $	QR: $Q_k \underline{T}_k = \begin{bmatrix} R_k \\ 0 \end{bmatrix}$	$x_k = \bar{V}_k y_k$ $\in \text{range}(\bar{V}_k)$
CS-MINRES-QLP	$y_k = \arg \min_{y \in \mathbb{C}^k} \ y\ $ s.t. $y \in \arg \min \ \underline{T}_k y - \beta_1 e_1\ $	QLP: $Q_k \underline{T}_k P_k = \begin{bmatrix} L_k \\ 0 \end{bmatrix}$	$x_k = \bar{V}_k y_k$ $\in \text{range}(\bar{V}_k)$
SH-MINRES	$y_k = \arg \min_{y \in \mathbb{R}^k} \ \underline{T}_k y - \beta_1 e_1\ $	QR: $Q_k \underline{T}_k = \begin{bmatrix} R_k \\ 0 \end{bmatrix}$	$x_k = \bar{V}_k y_k$ $\in \text{range}(\bar{V}_k)$
SH-MINRES-QLP	$y_k = \arg \min_{y \in \mathbb{R}^k} \ y\ $ s.t. $y \in \arg \min \ \underline{T}_k y - \beta_1 e_1\ $	QLP: $Q_k \underline{T}_k P_k = \begin{bmatrix} L_k \\ 0 \end{bmatrix}$	$x_k = \bar{V}_k y_k$ $\in \text{range}(\bar{V}_k)$

TABLE C.2

Bases, subproblem solutions, storage, and work for each method.

Method	New basis	z_k, t_k, u_k	x_k estimate	vecs	flops
cgLanczos	$W_k \equiv V_k L_k^{-T}$	$L_k D_k z_k = \beta_1 e_1$	$x_k = W_k z_k$	5	$8n$
SYMMLQ	$W_k \equiv V_{k+1} Q_k^T \begin{bmatrix} I_k \\ 0 \end{bmatrix}$	$L_k z_k = \beta_1 e_1$	$x_k = W_k z_k$	6	$9n$
MINRES	$D_k \equiv V_k R_k^{-1}$	$t_k = \beta_1 [I_k \ 0] Q_k e_1$	$x_k = D_k t_k$	7	$9n$
MINRES-QLP	$W_k \equiv V_k P_k$	$L_k u_k = \beta_1 [I_k \ 0] Q_k e_1$	$x_k = W_k u_k$	8	$14n$
CS-MINRES SH-MINRES	$D_k \equiv \bar{V}_k R_k^{-1}$	$t_k = \beta_1 [I_k \ 0] Q_k e_1$	$x_k = D_k t_k$	7	$9n$
CS-MINRES-QLP SH-MINRES-QLP	$W_k \equiv \bar{V}_k P_k$	$L_k u_k = \beta_1 [I_k \ 0] Q_k e_1$	$x_k = W_k u_k$	8	$14n$

REFERENCES

- [1] M. Arioli and S. Gratton. Least-squares problems, normal equations, and stopping criteria for the conjugate gradient method. Technical Report RAL-TR-2008-008, Rutherford Appleton Laboratory, Oxfordshire, UK, 2008.
- [2] A. Bunse-Gerstner and R. Stöver. On a conjugate gradient-type method for solving complex symmetric linear systems. *Linear Algebra Appl.*, 287(1-3):105–123, 1999. Special issue celebrating the 60th birthday of Ludwig Elsner.
- [3] R. H. Chan and X.-Q. Jin. Circulant and skew-circulant preconditioners for skew-Hermitian type Toeplitz systems. *BIT*, 31(4):632–646, 1991.
- [4] R. H.-F. Chan and X.-Q. Jin. *An Introduction to Iterative Toeplitz Solvers*. Society for Industrial and Applied Mathematics, 2007.
- [5] X.-W. Chang, C. C. Paige, and D. Titley-Péloquin. Stopping criteria for the iterative solution of linear least squares problems. *SIAM J. Matrix Anal. Appl.*, 31(2):831–852, 2009.
- [6] S.-C. Choi, D. L. Donoho, A. G. Flesia, X. Huo, O. Levi, and D. Shi. About Beamlab—a toolbox for new multiscale methodologies. <http://www-stat.stanford.edu/~beamlab/>, 2002.
- [7] S.-C. T. Choi. *Iterative Methods for Singular Linear Equations and Least-Squares Problems*. PhD thesis, ICME, Stanford University, 2006.
- [8] S.-C. T. Choi, C. C. Paige, and M. A. Saunders. MINRES-QLP: A Krylov subspace method for indefinite or singular symmetric systems. *SIAM J. Sci. Comput.*, 33(4):1810–1836, 2011.
- [9] S.-C. T. Choi and M. A. Saunders. ALGORITHM xxx: MINRES-QLP for singular symmetric and Hermitian linear equations and least-squares problems. *ACM Trans. Math. Software*. accepted Jan 2012.
- [10] S.-C. T. Choi and M. A. Saunders. MINRES-QLP MATLAB package. <http://code.google.com/p/minres-qlp/>, 2011.
- [11] S.-C. T. Choi and M. A. Saunders. ALGORITHM & DOCUMENTATION: MINRES-QLP for singular symmetric and Hermitian linear equations and least-squares problems. Technical Report ANL/MCS-P3027-0812, Computation Institute, University of Chicago, IL, 2012.
- [12] S.-C. T. Choi and M. A. Saunders. MINRES-QLP FORTRAN 90 package. <http://code.google.com/p/minres-qlp/>, 2012.
- [13] J. Claerbout. Hypertext documents about reproducible research. <http://sepwww.stanford.edu/doku.php?id=sep:research:reproducible>.
- [14] I. S. Duff. MA57—a code for the solution of sparse symmetric definite and indefinite systems. *ACM Trans. Math. Software*, 30(2):118–144, 2004.
- [15] R. Fletcher. Conjugate gradient methods for indefinite systems. In *Numerical Analysis (Proc 6th Biennial Dundee Conf., Univ. Dundee, Dundee, 1975)*, pages 73–89. Lecture Notes in Math., Vol. 506. Springer, Berlin, 1976.
- [16] L. Foster. SJSingular—MATLAB toolbox for managing the SJSU singular matrix collection. <http://www.math.sjsu.edu/singular/matrices/SJSingular.html>, 2008.
- [17] L. Foster. San Jose State University singular matrix database. <http://www.math.sjsu.edu/singular/matrices/>, 2009.
- [18] R. W. Freund. Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices. *SIAM J. Sci. Statist. Comput.*, 13(1):425–448, 1992.
- [19] R. W. Freund and N. M. Nachtigal. QMR: a quasi-minimal residual method for non-Hermitian linear systems. *Numer. Math.*, 60(3):315–339, 1991.
- [20] D. F. Gleich and L.-H. Lim. Rank aggregation via nuclear norm minimization. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 60–68, New York, NY, USA, 2011. ACM.
- [21] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, 3rd edition, 1996.
- [22] C. Greif and J. Varah. Iterative solution of skew-symmetric linear systems. *SIAM Journal on Matrix Analysis and Applications*, 31(2):584–601, 2009.
- [23] C. Guo and S. Qiao. A stable Lanczos tridiagonalization of complex symmetric matrices. Technical Report CAS 03-08-SQ, Department of Computing and Software, McMaster University, Ontario, Canada, 2003.
- [24] J. Hadamard. Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton University Bulletin*, XIII(4):49–52, 1902.
- [25] M. Hanke and J. G. Nagy. Restoration of atmospherically blurred images by symmetric indefinite conjugate gradient techniques. *Inverse Problems*, 12(2):157–173, 1996.
- [26] P. C. Hansen and D. P. O’Leary. The use of the L-curve in the regularization of discrete ill-posed problems. *SIAM J. Sci. Comput.*, 14(6):1487–1503, 1993.
- [27] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J.*

- Research Nat. Bur. Standards*, 49:409–436, 1952.
- [28] N. J. Higham. *Functions of matrices*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008. Theory and computation.
 - [29] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, Cambridge, 1990. Corrected reprint of the 1985 original.
 - [30] R. A. Horn and C. R. Johnson. *Topics in matrix analysis*. Cambridge University Press, Cambridge, 1991.
 - [31] D. A. Huckaby and T. F. Chan. On the convergence of Stewart’s QLP algorithm for approximating the SVD. *Numer. Algorithms*, 32(2-4):287–316, 2003.
 - [32] F. Incertis. A skew-symmetric formulation of the algebraic riccati equation problem. *Automatic Control, IEEE Transactions on*, 29(5):467–470, 1984.
 - [33] X. Jiang, L.-H. Lim, Y. Yao, and Y. Ye. Statistical ranking and combinatorial Hodge theory. *Math. Program.*, 127(1, Ser. B):203–244, 2011.
 - [34] P. Jiránek and D. Titley-Péloquin. Estimating the backward error in LSQR. *SIAM J. Matrix Anal. Appl.*, 31(4):2055–2074, 2010.
 - [35] M. Kilmer and G. W. Stewart. Iterative regularization and MINRES. *SIAM J. Matrix Anal. Appl.*, 21(2):613–628, 1999.
 - [36] C. Lanczos. *Applied analysis*. Englewood Cliffs, N.J., Prentice Hall, 1956.
 - [37] C. C. Paige. Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix. *J. Inst. Math. Appl.*, 18(3):341–349, 1976.
 - [38] C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.*, 12(4):617–629, 1975.
 - [39] C. C. Paige and M. A. Saunders. LSQR: an algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Software*, 8(1):43–71, 1982.
 - [40] 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, 1982.
 - [41] Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 7(3):856–869, 1986.
 - [42] M. A. Saunders. Computing projections with LSQR. *BIT*, 37(1):96–104, 1997.
 - [43] M. A. Saunders, H. D. Simon, and E. L. Yip. Two conjugate-gradient-type methods for unsymmetric linear equations. *SIAM Journal on Numerical Analysis*, 25(4):927–940, 1988.
 - [44] Systems Optimization Laboratory (SOL), Stanford University, downloadable software. <http://www.stanford.edu/group/SOL/software.html>.
 - [45] G. Stewart and J.-G. Sun. *Matrix perturbation theory*. Computer science and scientific computing. Academic Press, 1990.
 - [46] G. W. Stewart. On the continuity of the generalized inverse. *SIAM J. Appl. Math.*, 17:33–45, 1969.
 - [47] G. W. Stewart. Research, development and LINPACK. In J. R. Rice, editor, *Mathematical Software III*, pages 1–14. Academic Press, New York, 1977.
 - [48] G. W. Stewart. Updating a rank-revealing *ULV* decomposition. *SIAM J. Matrix Anal. Appl.*, 14(2):494–499, 1993.
 - [49] G. W. Stewart. The QLP approximation to the singular value decomposition. *SIAM J. Sci. Comput.*, 20(4):1336–1348, 1999.
 - [50] D. B. Szyld and O. B. Widlund. Variational analysis of some conjugate gradient methods. *East-West J. Numer. Math.*, 1(1):51–74, 1993.
 - [51] L. N. Trefethen and D. Bau, III. *Numerical Linear Algebra*. SIAM, Philadelphia, PA, 1997.
 - [52] H. A. Van der Vorst. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 13(2):631–644, 1992.
 - [53] O. Widlund. A lanczos method for a class of nonsymmetric systems of linear equations. *SIAM Journal on Numerical Analysis*, 15(4):801–812, 1978.

The submitted manuscript has been created by the University of Chicago as Operator of Argonne National Laboratory (“Argonne”) under Contract DE-AC02-06CH11357 with the U.S. Department of Energy. The U.S. Government retains for itself, and others acting on its behalf, a paid-up, nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.