A Tridiagonalization Method for Symmetric Saddle-Point Systems

Alfredo Buttari, Dominique Orban, Daniel Ruiz, David Titley-Peloquin

USYMLQR: A TRIDIAGONALIZATION METHOD FOR SYMMETRIC SADDLE-POINT SYSTEMS

ALFREDO BUTTARI*, DOMINIQUE ORBAN†, DANIEL RUIZ‡, AND DAVID TITLEY-PELOQUIN§

Abstract. We propose an iterative method named USYMLQR for the solution of symmetric saddle-point systems that exploits the orthogonal tridiagonalization method of Saunders, Simon, and Yip (1988). By contrast with methods based on the Golub and Kahan (1965) bidiagonalization process, our method takes advantage of two initial vectors and splits the system into the sum of a least-squares and a least-norm problem. In our numerical experiments, USYMLQR is competitive with and may require fewer operator-vector products than MINRES, yet performs a comparable amount of work per iteration and has comparable storage requirements.

Key words. Symmetric saddle-point systems, iterative methods, orthogonal tridiagonalization.

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

1. Introduction.

We consider the solution of symmetric saddle-point systems

$$\begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & \end{bmatrix} \begin{bmatrix} \mathbf{s} \\ \mathbf{t} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{c} \end{bmatrix}, \tag{1}$$

where $\mathbf{A}$ is m-by-n with $m \geqslant n$, and $\mathbf{M}$ is symmetric and positive definite. Such systems arise in numerous applications, including optimization, fluid dynamics, and data assimilation (Benzi, Golub, and Liesen, 2005). In the large-scale case, or the case where $\mathbf{M}$ and/or $\mathbf{A}$ is only available as an operator, it is common to employ a Krylov method to solve (1). Prime candidates are MINRES and SYMMLQ of Paige and Saunders (1975), both of which were designed with general symmetric indefinite systems in mind, but neither of which exploits the specific block structure of (1).

The main idea of this paper stems from the simple observation that any solution to (1) may be written as the sum of solutions of

$$\begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{c} \end{bmatrix}, \tag{2}$$

which are the optimality conditions of the least-squares and least-norm problems

$$\underset{\mathbf{x}}{\text{minimize}} \ \tfrac{1}{2}\|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2_{\mathbf{M}^{-1}} \quad \text{and} \quad \begin{array}{l} \underset{\mathbf{y}}{\text{minimize}} \ \tfrac{1}{2}\|\mathbf{y}\|^2_{\mathbf{M}} \\ \text{subject to } \mathbf{A}^T\mathbf{y} = \mathbf{c}, \end{array} \tag{3}$$

where the $\mathbf{V}$-norm of $\mathbf{p}$ is defined as $\|\mathbf{p}\|^2_{\mathbf{V}} := \mathbf{p}^T\mathbf{V}\mathbf{p}$ for any symmetric and positive definite $\mathbf{V}$. In the least-squares problem, we recover $\mathbf{r} = \mathbf{M}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x})$, while in the least-norm problem, we recover $\mathbf{z}$ as the (signed) Lagrange multipliers.

*Université de Toulouse, CNRS-IRIT, France. E-mail: alfredo.buttari@enseeiht.fr

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

‡Université de Toulouse, INPT-IRIT, France. E-mail: daniel.ruiz@enseeiht.fr

§Department of Bioresource Engineering, McGill University, Ste-Anne-de-Bellevue, QC, Canada. E-mail: david.titley-peloquin@mcgill.ca.

We propose an approach that meshes an iterative method for least-squares problems with one for least-norm problems in such a way that both problems (3) are solved in one pass. Each iteration of the proposed procedure has the same cost and almost the same storage requirements as one iteration of MINRES or SYMMLQ. In our numerical experiments, we have observed that our approach is competitive with and may solve (1) in fewer iterations than MINRES and SYMMLQ.

The two iterative methods are based on an orthogonal tridiagonalization process initially proposed by Saunders et al. (1988) for square, but not necessarily symmetric, matrices. This tridiagonalization process reduces to the symmetric Lanczos (1952) process when $\mathbf{A} = \mathbf{A}^T$ but differs from the Lanczos (1952) biorthogonalization process. By contrast with the Lanczos process, the Arnoldi (1951) process and the Golub and Kahan (1965) bidiagonalization, it must be initialized with two vectors $\mathbf{b}$ and $\mathbf{c}$. Saunders et al. (1988) note that, as a consequence, the tridiagonalization can be used to solve the pair of systems $\mathbf{Ax} = \mathbf{b}$ and $\mathbf{A}^T\mathbf{y} = \mathbf{c}$ at the same time. The resulting algorithms are named USYMQR and USYMLQ, respectively. Reichel and Ye (2008) remark that the process also applies with rectangular matrices $\mathbf{A}$, and that USYMQR can be used to solve least-squares problems, but only conduct numerical experiments on square systems.

Because our approach consists in transforming (1) to saddle-point systems with an identity block in place of $\mathbf{M}$, we do not discuss preconditioning issues in this paper. We assume that the user selected $\mathbf{M}$ so that it corresponds to a natural norm for measuring residuals and solutions. Applying further preconditioning would change those norms, and therefore, the problems in (2).

The remainder of the paper is organized as follows. We first establish that, for rectangular $\mathbf{A}$, USYMLQ solves a least-norm problem, and provide complete implementation details of both USYMQR and USYMLQ. We show how both methods mesh together to solve both problems of (3) in one pass. Although USYMQR and USYMLQ individually require more storage and have higher computational cost than methods based on the Golub and Kahan (1965) bidiagonalization, their combination yields a method with cost and storage comparable to that of MINRES or SYMMLQ applied to (1). Our numerical experiments show that our approach results in a similar overall number of operator-vector products as MINRES to decrease the residual by a comparable amount. The main difference is that we monitor convergence differently than in MINRES. We construct approximate solutions to (1) by exploiting the formulation and the related block structure of (2). In that respect, the strutured backward error analysis detailed in section 5 shows that monitoring the two sets of approximate solutions yields an acceptable solution to (4) provided that the blocks are not too ill conditioned. LSQR and CRAIG could be used to solve the two subproblems separately, whereas USYMLQR solves them concurrently. In section 7, we explain how to take the elliptic norms of (3) into account and relate USYMLQR to a block-Lanczos approach applied to (4).

Contributions. There are four main contributions: (i) we provide full implementation details on both USYMQR and USYMLQ for the simultaneous solution of (3) with $\mathbf{M} = \mathbf{I}$; (ii) we provide insight into the performance of the USYMQR/USYMLQ combination compared to MINRES applied to (1); (iii) we describe the solution of regularized problems; and (iv) we provide a variant of the orthogonalization process to general metrics $\mathbf{M}$.

Related Research. Reichel and Ye (2008) employ the orthogonal tridiagonalization process of Saunders et al. (1988) to derive a minimum-residual method for rectangular systems. Their method is named GLSQR, and is identical to the method

USYMQR proposed by Saunders et al. (1988) for square systems. However, all numerical experiments in (Reichel and Ye, 2008) are performed on square systems.

Golub, Stoll, and Wathen (2008) solve two square systems $\mathbf{A}\mathbf{x} = \mathbf{b}$ and $\mathbf{A}^T\mathbf{y} = \mathbf{c}$ in a scattering amplitude estimation application. Their approach consists in applying GLSQR twice, once to each system. They do not consider USYMLQ.

Orban and Arioli (2017) propose families of methods for systems of the form (4) that are also allowed to contain regularization. They all consist in first shifting the system to obtain a right-hand side with either $\mathbf{b} = \mathbf{0}$ or $\mathbf{c} = \mathbf{0}$, and subsequently shifting the solution. The shifted system is interpreted as a regularized least-squares problem in elliptic norms.

Notation. The notation $\mathbf{e}_i$ indicates the i-th canonical basis vector, and $\mathbf{I}_k$ is the k-by-k identity matrix. We use bold lowercase latin letter to denote full-space vectors and corresponding lightface letters to denote their expression in the basis of a Krylov-like subspace, e.g., $\mathbf{x} = \mathbf{Q}x$, with the exception of c_k and s_k, which denote a cosine and a sine participating in an orthogonal transformation. We use $\mathbf{0}$ to denote the zero (column) vector of appropriate size. All vectors are column vectors. For aesthetic reasons, we sometimes write a vector componentwise $x = (\xi_1, \ldots, \xi_n)$ in the text instead of $x = \begin{bmatrix}\xi_1 & \cdots & \xi_n\end{bmatrix}^T$.

2. Background and motivation.

Our approach can be motivated using

$$\begin{bmatrix} \mathbf{I} & \mathbf{A} \\ \mathbf{A}^T & \end{bmatrix} \begin{bmatrix} \mathbf{s} \\ \mathbf{t} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{c} \end{bmatrix}, \tag{4}$$

where we assume that $\mathbf{A}$ has full column rank so that (4) possesses a unique solution (Benzi et al., 2005, Theorem 3.1). In section 7, we describe modifications allowing changes in the metric used to measure $\mathbf{s}$ that yield a procedure for the solution of (1).

Saunders et al. (1988) introduce an iterative process to tridiagonalize a general square matrix by way of orthogonal transformations.

Algorithm 1 Saunders-Simon-Yip (SSY) Tridiagonalization

Require: $\mathbf{A}$, $\mathbf{b}$, $\mathbf{c}$

1: $\mathbf{u}_0 = \mathbf{0}$, $\mathbf{v}_0 = \mathbf{0}$
2: $\beta_1\mathbf{u}_1 = \mathbf{b}$, and $\gamma_1\mathbf{v}_1 = \mathbf{c}$, $(\beta_1, \gamma_1) > 0$ so that $\|\mathbf{u}_1\|_2 = \|\mathbf{v}_1\|_2 = 1$
3: **for** $k = 1, 2, \ldots$ **do**
4: $\quad \mathbf{q} = \mathbf{A}\mathbf{v}_k - \gamma_k\mathbf{u}_{k-1}$, $\alpha_k = \mathbf{u}_k^T\mathbf{q}$
5: $\quad \beta_{k+1}\mathbf{u}_{k+1} = \mathbf{q} - \alpha_k\mathbf{u}_k$, $\beta_{k+1} > 0$ so that $\|\mathbf{u}_{k+1}\|_2 = 1$
6: $\quad \gamma_{k+1}\mathbf{v}_{k+1} = \mathbf{A}^T\mathbf{u}_k - \beta_k\mathbf{v}_{k-1} - \alpha_k\mathbf{v}_k$, $\gamma_{k+1} > 0$ so that $\|\mathbf{v}_{k+1}\|_2 = 1$
7: **end for**

By the end of iteration k, Algorithm 1 has generated matrices $\mathbf{U}_k = \begin{bmatrix}\mathbf{u}_1 \ldots \mathbf{u}_k\end{bmatrix}$ and $\mathbf{V}_k = \begin{bmatrix}\mathbf{v}_1 \ldots \mathbf{v}_k\end{bmatrix}$ with theoretically orthonormal columns such that

$$\mathbf{A}\mathbf{V}_k = \mathbf{U}_k\mathbf{T}_k + \beta_{k+1}\mathbf{u}_{k+1}\mathbf{e}_k^T = \mathbf{U}_{k+1}\mathbf{T}_{k+1,k} \tag{5a}$$

$$\mathbf{A}^T\mathbf{U}_k = \mathbf{V}_k\mathbf{T}_k^T + \gamma_{k+1}\mathbf{v}_{k+1}\mathbf{e}_k^T = \mathbf{V}_{k+1}\mathbf{T}_{k,k+1}^T, \tag{5b}$$

where

$$\mathbf{T}_k = \begin{bmatrix} \alpha_1 & \gamma_2 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \gamma_k \\ & & \beta_k & \alpha_k \end{bmatrix}, \quad \mathbf{T}_{k+1,k} = \begin{bmatrix} \mathbf{T}_k \\ \beta_{k+1}\mathbf{e}_k^T \end{bmatrix}, \quad \mathbf{T}_{k,k+1} = \begin{bmatrix} \mathbf{T}_k & \gamma_{k+1}\mathbf{e}_k \end{bmatrix}.$$

In exact arithmetic, we have $\mathbf{U}_k^T \mathbf{A} \mathbf{V}_k = \mathbf{T}_k$, so that after n iterations, singular values are preserved in exact arithmetic. Note that (5) differs from the outcome of the Lanczos (1952) biorthogonalization process for square matrices $\mathbf{A}$, which also produces tridiagonal $\mathbf{T}_k$ but theoretically biorthogonal $\mathbf{W}_k$ and $\mathbf{Y}_k$ such that $\mathbf{W}_k^T \mathbf{Y}_k = \mathbf{I}$, $\mathbf{Y}_k^T \mathbf{A} \mathbf{Y}_k = \mathbf{T}_k$ and $\mathbf{W}_k^T \mathbf{A}^T \mathbf{W}_k = \mathbf{T}_k^T$, so that eigenvalues, not singular values, are preserved after n iterations. Contrary to the biorthogonalization process, Algorithm 1 equally applies to rectangular matrices.

An approach to solving the least-squares problem in (3) is to seek $\mathbf{x}_k = \mathbf{V}_k x_k$ and select x_k so as to minimize the norm of the residual $\mathbf{b} - \mathbf{A}\mathbf{x}_k = \mathbf{U}_{k+1}(\beta_1 \mathbf{e}_1 - \mathbf{T}_{k+1,k} x_k)$. Because $\mathbf{U}_{k+1}$ has orthonormal columns, this means finding $x_k \in \mathbb{R}^k$ as a solution of

$$\underset{x}{\text{minimize}} \ \|\beta_1 \mathbf{e}_1 - \mathbf{T}_{k+1,k} x\|. \tag{6}$$

To compute an approximate solution of the least-norm problem in (3), we seek $\mathbf{y}_k^L = \mathbf{U}_{k+1} y_k^L$ where $y_k^L \in \mathbb{R}^{k+1}$ solves

$$\underset{y}{\text{minimize}} \ \|y\| \quad \text{subject to} \ \mathbf{T}_{k+1,k}^T y = \gamma_1 \mathbf{e}_1, \tag{7}$$

see (Saunders et al., 1988, §5). If $\mathbf{A}^T \mathbf{y} = \mathbf{c}$ is compatible, (7) possesses a unique solution, even though $\mathbf{T}_k$ could be singular.

If one could guarantee that $\mathbf{T}_k$ is nonsingular, it would be possible to devise a conjugate-gradient-type method that seeks approximate solutions $\mathbf{x}_k^C := \mathbf{V}_k x_k^C$ and $\mathbf{y}_k^C = \mathbf{U}_k y_k^C$ where $x_k^C \in \mathbb{R}^k$ and $y_k^C \in \mathbb{R}^k$ are found by imposing the Galerkin conditions $\mathbf{U}_k^T(\mathbf{b} - \mathbf{A}\mathbf{x}_k^C) = \mathbf{0}$ and $\mathbf{V}_k^T(\mathbf{c} - \mathbf{A}^T \mathbf{y}_k^C) = \mathbf{0}$. Introducing the definition of $\mathbf{x}_k^C$ and $\mathbf{y}_k^C$ into (5a) and (5b), we obtain the tridiagonal systems

$$\mathbf{T}_k x_k^C = \beta_1 \mathbf{e}_1 \qquad \text{and} \qquad \mathbf{T}_k^T y_k^C = \gamma_1 \mathbf{e}_1. \tag{8}$$

Saunders et al. (1988) call the methods defined by (6) and (7) USYMQR and USYMLQ, respectively. When $\mathbf{A}$ is square and symmetric, USYMQR and USYMLQ coincide with MINRES and SYMMLQ of Paige and Saunders (1975), respectively, and the method based on (8) coincides with the conjugate gradient method.

USYMQR is referred to as GLSQR by Reichel and Ye (2008) and Golub et al. (2008), though it does not reduce to LSQR (Paige and Saunders, 1982) in any particular case.

For all $i, j \geqslant 0$, the vectors $\mathbf{u}_i$ and $\mathbf{v}_j$ satisfy

$$\mathbf{u}_{2i} \in \text{Span}\{\mathbf{b}, (\mathbf{A}\mathbf{A}^T)\mathbf{b}, \ldots, (\mathbf{A}\mathbf{A}^T)^{i-1}\mathbf{b}, \mathbf{A}\mathbf{c}, \ldots, (\mathbf{A}\mathbf{A}^T)^{i-1}\mathbf{A}\mathbf{c}\}, \tag{9a}$$

$$\mathbf{u}_{2i+1} \in \text{Span}\{\mathbf{b}, (\mathbf{A}\mathbf{A}^T)\mathbf{b}, \ldots, (\mathbf{A}\mathbf{A}^T)^{i}\mathbf{b}, \mathbf{A}\mathbf{c}, \ldots, (\mathbf{A}\mathbf{A}^T)^{i-1}\mathbf{A}\mathbf{c}\}, \tag{9b}$$

$$\mathbf{v}_{2j} \in \text{Span}\{\mathbf{c}, (\mathbf{A}^T\mathbf{A})\mathbf{c}, \ldots, (\mathbf{A}^T\mathbf{A})^{j-1}\mathbf{c}, \mathbf{A}^T\mathbf{b}, \ldots, (\mathbf{A}^T\mathbf{A})^{j-1}\mathbf{A}^T\mathbf{b}\}, \tag{9c}$$

$$\mathbf{v}_{2j+1} \in \text{Span}\{\mathbf{c}, (\mathbf{A}^T\mathbf{A})\mathbf{c}, \ldots, (\mathbf{A}^T\mathbf{A})^{j}\mathbf{c}, \mathbf{A}^T\mathbf{b}, \ldots, (\mathbf{A}^T\mathbf{A})^{j-1}\mathbf{A}^T\mathbf{b}\}. \tag{9d}$$

Methods based on the Golub and Kahan (1965) process such as LSQR (Paige and Saunders, 1982) and CRAIG (Craig, 1955) use a single starting vector but, much like Algorithm 1, build left and right orthonormal bases. LSQR is appropriate for the least-squares problem in (3) and can be initialized with $\mathbf{b}$. It generates left and right orthonormal vectors that form a basis for $\text{Span}\{\mathbf{b}, (\mathbf{A}\mathbf{A}^T)\mathbf{b}, \ldots, (\mathbf{A}\mathbf{A}^T)^{i-1}\mathbf{b}\}$ and $\text{Span}\{\mathbf{A}^T\mathbf{b}, (\mathbf{A}^T\mathbf{A})\mathbf{A}^T\mathbf{b}, \ldots, (\mathbf{A}^T\mathbf{A})^{i-1}\mathbf{A}^T\mathbf{b}\}$, respectively. Similarly, CRAIG is appropriate for the least-norm problem in (3) and can be initialized with $\mathbf{c}$. It generates the left and right orthonormal vectors that form a basis for $\text{Span}\{\mathbf{A}\mathbf{c}, \ldots, (\mathbf{A}\mathbf{A}^T)^{j-1}\mathbf{A}\mathbf{c}\}$ and $\text{Span}\{\mathbf{c}, (\mathbf{A}^T\mathbf{A})\mathbf{c}, \ldots, (\mathbf{A}^T\mathbf{A})^{j-1}\mathbf{c}\}$, respectively. Thus Algorithm 1 can be interpreted as interleaving and orthogonalizing the LSQR and CRAIG orthogonal sequences.

Regarding the solution of (1), our only assumption is that $\mathbf{c} \in \text{Range}(\mathbf{A}^T)$. Under this assumption, both problems in (3) are feasible, so that both systems in (2) are consistent, and so is (1).

The following property states that USYMQR applied to rank-deficient least-squares problems identifies the minimum least-squares solution. The proof is similar to that of (Fong and Saunders, 2011, Theorem 4.2).

THEOREM 1. *If* $\mathbf{c} \in \text{Range}(\mathbf{A}^T)$, USYMQR *finds the minimum-norm solution of the least-squares problem in* (3).

Proof. Any solution $\mathbf{x}$ of the least-squares problem in (3) with $\mathbf{M} = \mathbf{I}$ satisfies $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$. Let $\mathbf{x}_\star$ be the solution identified by USYMQR, $\bar{\mathbf{x}}$ be another solution and $\mathbf{d} := \bar{\mathbf{x}} - \mathbf{x}_\star$. Then, $\mathbf{A}^T\mathbf{A}\mathbf{d} = \mathbf{0}$ and thus, $\mathbf{A}\mathbf{d} = \mathbf{0}$. By construction, there exists k such that $\mathbf{x}_\star \in \text{Range}(\mathbf{V}_k)$, i.e., there exists $x_\star \in \mathbb{R}^k$ such that $\mathbf{x}_\star = \mathbf{V}_k x_\star$. For all $j \geqslant 0$, (9c)–(9d) are satisfied and only $\mathbf{v}_1$ has a component along $\mathbf{c}$. Thus, $\mathbf{d}^T\mathbf{x}_\star = \mathbf{d}^T\mathbf{V}_k x_\star = \mathbf{d}^T\mathbf{v}_1\xi_1$, where ξ_1 is the first component of $x_\star$. However, our assumption that $\mathbf{c} \in \text{Range}(\mathbf{A}^T)$ implies $\mathbf{d}^T\mathbf{c} = 0$ and therefore $\mathbf{d}^T\mathbf{x}_\star = 0$. Consequently,

$$\|\bar{\mathbf{x}}\|^2 - \|\mathbf{x}_\star\|^2 = \|\mathbf{x}_\star + \mathbf{d}\|^2 - \|\mathbf{x}_\star\|^2 = \|\mathbf{d}\|^2 + 2\mathbf{d}^T\mathbf{x}_\star = \|\mathbf{d}\|^2 \geqslant 0,$$

and $\mathbf{x}_\star$ is the minimum-norm least-squares solution. ∎

3. Implementation. In this section, we give complete implementation details of USYMQR and USYMLQ for the solution of (4). We begin with USYMQR, and then explain how it meshes with USYMLQ in order to solve both problems of (3) at once. This will put us in good position to explain how to take ellipsoidal norms into account at minimal cost.

3.1. Least-squares subproblem: USYMQR iteration. In this section, we focus on the problem

$$\begin{bmatrix} \mathbf{I} & \mathbf{A} \\ \mathbf{A}^T & \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \quad \Longleftrightarrow \quad \underset{\mathbf{x}}{\text{minimize}} \ \tfrac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2, \tag{10}$$

and initialize Algorithm 1 with $\mathbf{A}$, $\mathbf{b}$ and $\mathbf{c}$.

3.1.1. Solution update. The subproblem solved at each iteration of USYMQR is the overdetermined linear least-squares problem (6). The solution x_k can be obtained

via the QR factorization

$$\mathbf{T}_{k+1,k} = \mathbf{Q}_{k+1} \begin{bmatrix} \mathbf{R}_k \\ \mathbf{0}^{\mathsf{T}} \end{bmatrix}, \qquad \mathbf{R}_k := \begin{bmatrix} \delta_1 & \lambda_1 & \epsilon_1 & & \\ & \delta_2 & \lambda_2 & \ddots & \\ & & \delta_3 & \ddots & \epsilon_{k-2} \\ & & & \ddots & \lambda_{k-1} \\ & & & & \delta_k \end{bmatrix}, \tag{11}$$

where $\mathbf{Q}_{k+1}^{\mathsf{T}} = \mathbf{Q}_{k,k+1}\mathbf{Q}_{k-1,k}\dots\mathbf{Q}_{1,2}$ is a product of reflections, and $\mathbf{R}_k$ is upper triangular with three nonzero diagonals. Then x_k is found as the solution of

$$\underset{x}{\text{minimize}} \left\| \mathbf{Q}_{k+1}^{\mathsf{T}}(\beta_1 \mathbf{e}_1) - \begin{bmatrix} \mathbf{R}_k \\ \mathbf{0}^{\mathsf{T}} \end{bmatrix} x \right\|. \tag{12}$$

The subdiagonals of $\mathbf{T}_{k+1,k}$ can be zeroed out by premultiplying with reflections, each of which affects two rows and three columns. The k-th reflection $\mathbf{Q}_{k,k+1}$ can be represented as

$$\begin{array}{c} \\ k \\ k+1 \end{array} \begin{array}{c} \begin{array}{cc} k & k+1 \end{array} \\ \begin{bmatrix} c_k & s_k \\ s_k & -c_k \end{bmatrix} \end{array} \begin{array}{c} \begin{array}{ccc} k & k+1 & k+2 \end{array} \\ \begin{bmatrix} \bar{\delta}_k & \bar{\lambda}_k & \\ \beta_{k+1} & \alpha_{k+1} & \gamma_{k+2} \end{bmatrix} \end{array} = \begin{array}{c} \begin{array}{ccc} k & k+1 & k+2 \end{array} \\ \begin{bmatrix} \delta_k & \lambda_k & \epsilon_k \\ 0 & \bar{\delta}_{k+1} & \bar{\lambda}_{k+1} \end{bmatrix} \end{array}, \tag{13}$$

where elements decorated by a bar are to be updated by the next reflection, and the border indices are row and column indices. For the purpose of establishing recursion formulae, we define $\bar{\delta}_1 := \alpha_1$ and $\bar{\lambda}_1 := \gamma_2$. The k-th reflection zeros out β_{k+1}, i.e.,

$$\delta_k := \left(\bar{\delta}_k^2 + \beta_{k+1}^2\right)^{\frac{1}{2}}, \quad c_k := \bar{\delta}_k/\delta_k, \quad s_k := \beta_{k+1}/\delta_k. \tag{14}$$

We then have the recursion formulae

$$\lambda_k = c_k\bar{\lambda}_k + s_k\alpha_{k+1}, \quad \bar{\delta}_{k+1} = s_k\bar{\lambda}_k - c_k\alpha_{k+1}, \tag{15}$$

$$\epsilon_k = s_k\gamma_{k+2}, \qquad \bar{\lambda}_{k+1} = -c_k\gamma_{k+2}. \tag{16}$$

The effect of $\mathbf{Q}_{k+1}^{\mathsf{T}}$ on the right-hand side $\beta_1\mathbf{e}_1$ may be described as

$$\begin{array}{c} \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{bmatrix} c_1 & s_1 \\ s_1 & -c_1 \end{bmatrix} \end{array} \begin{bmatrix} \beta_1 \\ 0 \end{bmatrix} = \begin{bmatrix} \phi_1 \\ \bar{\phi}_2 \end{bmatrix}, \qquad \begin{array}{c} \\ k \\ k+1 \end{array} \begin{array}{c} \begin{array}{cc} k & k+1 \end{array} \\ \begin{bmatrix} c_k & s_k \\ s_k & -c_k \end{bmatrix} \end{array} \begin{bmatrix} \bar{\phi}_k \\ 0 \end{bmatrix} = \begin{bmatrix} \phi_k \\ \bar{\phi}_{k+1} \end{bmatrix}, \tag{17}$$

with

$$\bar{\phi}_1 := \beta_1, \quad \text{and} \quad \phi_k = c_k\bar{\phi}_k, \quad \bar{\phi}_{k+1} = s_k\bar{\phi}_k, \quad k = 1, 2, \dots$$

Let $f_k := (\phi_1, \dots, \phi_k)$ and $\bar{f}_{k+1} := \mathbf{Q}_{k+1}^{\mathsf{T}}(\beta_1\mathbf{e}_1) = (\phi_1, \dots, \phi_k, \bar{\phi}_{k+1}) = (f_k, \bar{\phi}_{k+1})$. Then, the solution of (6) is $x_k = \mathbf{R}_k^{-1}f_k$, and the transformed residual is

$$\mathbf{Q}_{k+1}^{\mathsf{T}}(\beta_1\mathbf{e}_1) - \begin{bmatrix} \mathbf{R}_k \\ \mathbf{0}^{\mathsf{T}} \end{bmatrix} x_k = \bar{\phi}_{k+1}\mathbf{e}_{k+1}. \tag{18}$$

Because $\mathbf{R}_k$ is upper triangular, the entire vector x_k likely changes at each iteration. Fortunately, we may update $\mathbf{x}_k$ directly instead as in (Paige and Saunders, 1975, Equation (4.3)). Indeed,

$$\mathbf{x}_k = \mathbf{V}_k x_k = \mathbf{V}_k\mathbf{R}_k^{-1}f_k = \mathbf{W}_k f_k, \qquad \mathbf{W}_k := \mathbf{V}_k\mathbf{R}_k^{-1}. \tag{19}$$

If $\mathbf{w}_j$ denotes the j-th column of $\mathbf{W}_k$, the identity $\mathbf{R}_k^{\mathsf{T}}\mathbf{W}_k^{\mathsf{T}} = \mathbf{V}_k^{\mathsf{T}}$ yields the recursion

$$\mathbf{w}_1 := \frac{\mathbf{v}_1}{\delta_1}, \quad \mathbf{w}_2 = \frac{\mathbf{v}_2 - \lambda_1\mathbf{w}_1}{\delta_2}, \quad \mathbf{w}_k = \frac{\mathbf{v}_k - \lambda_{k-1}\mathbf{w}_{k-1} - \epsilon_{k-2}\mathbf{w}_{k-2}}{\delta_k}, \quad k \geqslant 3.$$

In turn, this gives the update $\mathbf{x}_k = \mathbf{x}_{k-1} + \phi_k\mathbf{w}_k$.

3.1.2. Residuals. We have from (5a), (11) and (18)

$$\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k = \mathbf{U}_{k+1}(\beta_1\mathbf{e}_1 - \mathbf{T}_{k+1,k}x_k) = \bar{\phi}_{k+1}\mathbf{U}_{k+1}\mathbf{Q}_{k+1}\mathbf{e}_{k+1} = \mathbf{U}_{k+1}r_k. \tag{20}$$

Thus,

$$\|\mathbf{r}_k\| = |\bar{\phi}_{k+1}| = |s_k\bar{\phi}_k| = \cdots = |s_k s_{k-1}\dots s_1|\beta_1. \tag{21}$$

From the above expression, it is clear that the residual norm is non-increasing. Note that a simple recursion for $\mathbf{r}_k$ is available in case the residual vector is required. The definitions of $\mathbf{U}_{k+1}$ and $\mathbf{Q}_{k+1}$ together with (20) yield

$$\begin{aligned}
\mathbf{r}_k &= \bar{\phi}_{k+1}\begin{bmatrix}\mathbf{U}_k & \mathbf{u}_{k+1}\end{bmatrix}\begin{bmatrix}\mathbf{Q}_k & \\ & 1\end{bmatrix}\mathbf{Q}_{k,k+1}^{\mathsf{T}}\mathbf{e}_{k+1} \\
&= s_k\bar{\phi}_k\begin{bmatrix}\mathbf{U}_k & \mathbf{u}_{k+1}\end{bmatrix}\begin{bmatrix}\mathbf{Q}_k & \\ & 1\end{bmatrix}(s_k\mathbf{e}_k - c_k\mathbf{e}_{k+1}) \\
&= s_k^2\bar{\phi}_k\mathbf{U}_k\mathbf{Q}_k\mathbf{e}_k - s_k c_k\bar{\phi}_k\mathbf{u}_{k+1} \\
&= s_k^2\mathbf{r}_{k-1} - c_k\bar{\phi}_{k+1}\mathbf{u}_{k+1}.
\end{aligned}$$

Note that (11) and (20) together imply $\mathbf{T}_{k+1,k}^{\mathsf{T}}r_k = \bar{\phi}_{k+1}\mathbf{T}_{k+1,k}^{\mathsf{T}}\mathbf{Q}_{k+1}\mathbf{e}_{k+1} = \mathbf{0}$. In effect, we have approximated the solution of (10) with that of

$$\begin{bmatrix}\mathbf{I}_{k+1} & \mathbf{T}_{k+1,k} \\ \mathbf{T}_{k+1,k}^{\mathsf{T}} & \mathbf{0}_k\end{bmatrix}\begin{bmatrix}r_k \\ x_k\end{bmatrix} = \begin{bmatrix}\beta_1\mathbf{e}_1 \\ \mathbf{0}\end{bmatrix}.$$

Finally we need an expression for the optimality residual $\|\mathbf{A}^{\mathsf{T}}\mathbf{r}_k\|$ of the least-squares problem in (10). The expression (20) combines with (5b) to yield

$$\mathbf{A}^{\mathsf{T}}\mathbf{r}_k = \bar{\phi}_{k+1}\mathbf{A}^{\mathsf{T}}\mathbf{U}_{k+1}\mathbf{Q}_{k+1}\mathbf{e}_{k+1} = \bar{\phi}_{k+1}\mathbf{V}_{k+2}\mathbf{T}_{k+1,k+2}^{\mathsf{T}}\mathbf{Q}_{k+1}\mathbf{e}_{k+1}.$$

But

$$\mathbf{T}_{k+1,k+2}^{\mathsf{T}} = \begin{bmatrix}\mathbf{T}_{k+1}^{\mathsf{T}} \\ \gamma_{k+2}\mathbf{e}_{k+1}^{\mathsf{T}}\end{bmatrix} = \begin{bmatrix}\mathbf{T}_k^{\mathsf{T}} & \beta_{k+1}\mathbf{e}_k \\ \gamma_{k+1}\mathbf{e}_k^{\mathsf{T}} & \alpha_{k+1} \\ 0 & \gamma_{k+2}\end{bmatrix} = \begin{bmatrix}\mathbf{T}_{k+1,k}^{\mathsf{T}} \\ \gamma_{k+1}\mathbf{e}_k^{\mathsf{T}} + \alpha_{k+1}\mathbf{e}_{k+1}^{\mathsf{T}} \\ \gamma_{k+2}\mathbf{e}_{k+1}^{\mathsf{T}}\end{bmatrix},$$

so that

$$\mathbf{T}_{k+1,k+2}^{\mathsf{T}}\mathbf{Q}_{k+1}\mathbf{e}_{k+1} = \begin{bmatrix}\begin{bmatrix}\mathbf{R}_k^{\mathsf{T}} & \mathbf{0}\end{bmatrix} \\ \gamma_{k+1}\mathbf{e}_k^{\mathsf{T}}\mathbf{Q}_{k+1} + \alpha_{k+1}\mathbf{e}_{k+1}^{\mathsf{T}}\mathbf{Q}_{k+1} \\ \gamma_{k+2}\mathbf{e}_{k+1}^{\mathsf{T}}\mathbf{Q}_{k+1}\end{bmatrix}\mathbf{e}_{k+1}.$$

It is not difficult to verify that $\mathbf{e}_k^{\mathsf{T}}\mathbf{Q}_{k+1}\mathbf{e}_{k+1} = -c_{k-1}s_k$ and $\mathbf{e}_{k+1}^{\mathsf{T}}\mathbf{Q}_{k+1}\mathbf{e}_{k+1} = -c_k$, and therefore,

$$\mathbf{T}_{k+1,k+2}^{\mathsf{T}}\mathbf{Q}_{k+1}\mathbf{e}_{k+1} = \begin{bmatrix}\mathbf{0} \\ -c_{k-1}s_k\gamma_{k+1} - c_k\alpha_{k+1} \\ -c_k\gamma_{k+2}\end{bmatrix} = \begin{bmatrix}\mathbf{0} \\ s_k\bar{\lambda}_k - c_k\alpha_{k+1} \\ \bar{\lambda}_{k+1}\end{bmatrix} = \begin{bmatrix}\mathbf{0} \\ \bar{\delta}_{k+1} \\ \bar{\lambda}_{k+1}\end{bmatrix}.$$

Finally,

$$\mathbf{A}^T\mathbf{r}_k = \bar{\phi}_{k+1}(\bar{\delta}_{k+1}\mathbf{v}_{k+1} + \bar{\lambda}_{k+1}\mathbf{v}_{k+2}), \tag{22}$$

and by orthogonality,

$$\|\mathbf{A}^T\mathbf{r}_k\| = |\bar{\phi}_{k+1}|\sqrt{\bar{\delta}_{k+1}^2 + \bar{\lambda}_{k+1}^2}, \tag{23}$$

which is readily available.

3.2. Least-norm subproblem: USYMLQ iteration. We consider

$$\begin{bmatrix} \mathbf{I} & \mathbf{A} \\ \mathbf{A}^T & \end{bmatrix}\begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{c} \end{bmatrix} \quad \Longleftrightarrow \quad \underset{\mathbf{y}}{\text{minimize}} \ \tfrac{1}{2}\|\mathbf{y}\|^2 \quad \text{subject to } \mathbf{A}^T\mathbf{y} = \mathbf{c}, \tag{24}$$

and Algorithm 1 initialized with $\mathbf{A}$, $\mathbf{b}$ and $\mathbf{c}$ as in subsection 3.1.

3.2.1. Solution update. We use the factorization of $\mathbf{T}_{k+1,k}$ to update an approximate solution of the adjoint system $\mathbf{A}^T\mathbf{y} = \mathbf{c}$ during the USYMQR iterations. Indeed, we now seek an approximation of the form $\mathbf{y}_k := \mathbf{U}_{k+1}y_k$ as a solution to the least-norm problem in (3). After the reflection $\mathbf{Q}_{k,k+1}$, we have the LQ factorizations

$$\mathbf{T}_{k+1,k}^T\mathbf{Q}_{k+1} = \begin{bmatrix} \mathbf{R}_k^T & \mathbf{0} \end{bmatrix} \qquad \text{and} \qquad \mathbf{T}_{k+1}^T = \bar{\mathbf{R}}_{k+1}^T\mathbf{Q}_{k+1}^T, \tag{25}$$

where $\bar{\mathbf{R}}_{k+1}$ differs from $\mathbf{R}_{k+1}$ only in the $(k+1,k+1)$-th element, denoted $\bar{\delta}_{k+1}$. This factorization allows us to rewrite the constraints of (7) as

$$\mathbf{R}_k^T h_{k-1} = \gamma_1\mathbf{e}_1, \quad h_k := \mathbf{Q}_{k+1}^T y_k = (h_{k-1}, \eta_k) \in \mathbb{R}^{k+1}. \tag{26}$$

Because $\mathbf{R}_k^T$ is lower triangular, we obtain an update for $h_{k-1} = (\eta_1, \ldots, \eta_{k-1})$:

$$\eta_1 = \gamma_1/\delta_1, \quad \eta_2 = -\lambda_1\eta_1/\delta_2, \quad \eta_k = -(\lambda_{k-1}\eta_{k-1} + \epsilon_{k-2}\eta_{k-2})/\delta_k, \quad k \geqslant 3, \tag{27}$$

so that the solution of (7) is $y_k^L = \mathbf{Q}_{k+1}(h_{k-1}, 0)$. Similarly, $y_{k+1}^C = \mathbf{Q}_{k+1}\bar{h}_k$ where $\bar{h}_{k+1} = (h_{k-1}, \bar{\eta}_{k+1})$ with

$$\bar{\eta}_{k+1} = -(\lambda_k\eta_k + \epsilon_{k-1}\eta_{k-1})/\bar{\delta}_{k+1}$$

solves the second system of (8) at iteration $k+1$. Each $\bar{\eta}_j$ is updated to $\eta_j = \bar{\eta}_j\bar{\delta}_j/\delta_j = c_j\bar{\eta}_j$ when δ_j becomes available. As in SYMMLQ (Paige and Saunders, 1975), $\delta_k > \bar{\delta}_k$ so long as $\beta_{k+1} \neq 0$, so that $\mathbf{R}_k$ should be better conditioned than $\bar{\mathbf{R}}_k$ and the computed h_k should be more accurate than the computed $\bar{h}_k$. Both $\mathbf{y}_k^L := \mathbf{U}_{k+1}y_k^L$ and $\mathbf{y}_{k+1}^C := \mathbf{U}_{k+1}y_{k+1}^C$ may be updated efficiently once we define

$$\bar{\mathbf{P}}_{k+1} := \mathbf{U}_{k+1}\mathbf{Q}_{k+1} = \begin{bmatrix} \mathbf{p}_1 & \cdots & \mathbf{p}_k & \bar{\mathbf{p}}_{k+1} \end{bmatrix},$$

using the recursions

$$\begin{aligned} \mathbf{y}_k^L &= \mathbf{y}_{k-1}^L + \eta_k\mathbf{p}_k \\ \mathbf{y}_{k+1}^C &= \mathbf{y}_k^L + \bar{\eta}_{k+1}\bar{\mathbf{p}}_{k+1} \\ \mathbf{p}_{k+1} &= c_{k+1}\bar{\mathbf{p}}_{k+1} + s_{k+1}\mathbf{u}_{k+2} \\ \bar{\mathbf{p}}_{k+2} &= s_{k+1}\bar{\mathbf{p}}_{k+1} - c_{k+1}\mathbf{u}_{k+2}. \end{aligned}$$

The recursions are initialized with $\mathbf{y}_0^L := \mathbf{0}$, $\bar{\mathbf{p}}_1 := \mathbf{u}_1$, and $\mathbf{y}_1^C := \bar{\eta}_1\bar{\mathbf{p}}_1$.

3.2.2. Residuals. The residual at $\mathbf{y}_k^L = \mathbf{U}_{k+1} y_k^L$ or $\mathbf{y}_{k+1}^C = \mathbf{U}_{k+1} y_{k+1}^C$ is

$$\begin{aligned} \mathbf{r}_{k+1} &:= \mathbf{c} - \mathbf{A}^\mathsf{T}\mathbf{y} \\ &= \gamma_1 \mathbf{v}_1 - \mathbf{V}_{k+1}\mathbf{T}_{k+1}^\mathsf{T} y - \gamma_{k+2}\mathbf{v}_{k+2}\mathbf{e}_{k+1}^\mathsf{T} y \\ &= \mathbf{V}_{k+1}(\gamma_1 \mathbf{e}_1 - \mathbf{T}_{k+1}^\mathsf{T} y) - \gamma_{k+2}\mathbf{v}_{k+2}\mathbf{e}_{k+1}^\mathsf{T} y. \end{aligned} \tag{28}$$

The definition of $\mathbf{T}_k$ yields

$$\mathbf{T}_{k+1}^\mathsf{T} y = \begin{bmatrix} \gamma_1 \mathbf{e}_1 \\ \gamma_{k+1}\mathbf{e}_k^\mathsf{T} y + \alpha_{k+1}\mathbf{e}_{k+1}^\mathsf{T} y \end{bmatrix}.$$

Neither y_k^L nor y_{k+1}^C is directly available, but because

$$y_k^L = \mathbf{Q}_{k+1} \begin{bmatrix} h_{k-1} \\ 0 \end{bmatrix},$$

we have $\mathbf{e}_k^\mathsf{T} y_k^L = s_{k-1}\eta_{k-1} - c_{k-1}c_k\eta_k$ and $\mathbf{e}_{k+1}^\mathsf{T} y_k^L = s_k\eta_k$. Similarly, because $y_{k+1}^C = \mathbf{Q}_{k+1}^\mathsf{T}\bar{h}_{k+1}$, we obtain $\theta_{k+1} := \mathbf{e}_{k+1}^\mathsf{T} y_{k+1}^C = s_k\eta_k - c_k\bar{\eta}_{k+1}$, by identification.

The residual associated to $\mathbf{y}_k^L$ is then

$$\begin{aligned} \mathbf{r}_k^L &= -(\gamma_{k+1}(s_{k-1}\eta_{k-1} - c_{k-1}c_k\eta_k) + \alpha_{k+1}s_k\eta_k)\mathbf{v}_{k+1} - \gamma_{k+2}s_k\eta_k\mathbf{v}_{k+2} \\ &= -(\epsilon_{k-1}\eta_{k-1} + \lambda_k\eta_k)\mathbf{v}_{k+1} - \epsilon_k\eta_k\mathbf{v}_{k+2} \\ &= -\delta_{k+1}\eta_{k+1}\mathbf{v}_{k+1} - \epsilon_k\eta_k\mathbf{v}_{k+2}, \end{aligned} \tag{29}$$

where we used the recursions (15) and (27). By orthogonality,

$$\|\mathbf{r}_k^L\|^2 = (\delta_{k+1}\eta_{k+1})^2 + (\epsilon_k\eta_k)^2.$$

The residual associated to $\mathbf{y}_{k+1}^C = \mathbf{U}_{k+1} y_{k+1}^C$ is simpler to calculate because (8) and (28) directly imply

$$\mathbf{r}_{k+1}^C := \mathbf{V}_{k+1}(\gamma_1\mathbf{e}_1 - \mathbf{T}_{k+1}^\mathsf{T} y_{k+1}^C) - \gamma_{k+2}\mathbf{v}_{k+2}\mathbf{e}_{k+1}^\mathsf{T} y_{k+1}^C = -\gamma_{k+2}\theta_{k+1}\mathbf{v}_{k+2}. \tag{30}$$

Because $\mathbf{v}_{k+2}$ is a unit vector,

$$\|\mathbf{r}_{k+1}^C\| = \gamma_{k+2}\,|\theta_{k+1}|.$$

3.2.3. Computation of z. There remains to determine a recursion for $\mathbf{z}$ such that $\mathbf{Az} = -\mathbf{y}$ in (24). In view of (5a), because $\mathbf{y}$ must lie in the range of $\mathbf{A}$, we seek approximations $\mathbf{z}_k = \mathbf{V}_k z_k$, and note that

$$\mathbf{A}\mathbf{z}_k = \mathbf{A}\mathbf{V}_k z_k = \mathbf{U}_{k+1}\mathbf{T}_{k+1,k} z_k = -\mathbf{U}_{k+1} y_k.$$

Premultiplying both sides of the last equality by $\mathbf{U}_{k+1}^\mathsf{T}$ yields the subproblem

$$\mathbf{T}_{k+1,k} z_k = -y_k. \tag{31}$$

We premultiply with $\mathbf{Q}_k^\mathsf{T}$ and use the QR factorization (11), to obtain

$$\begin{bmatrix} \mathbf{R}_k \\ \mathbf{0}^\mathsf{T} \end{bmatrix} z_k = -h_k,$$

which is a situation similar to (19). Thus, because $h_k = (h_{k-1}, 0)$ for y_k^L, we may define z_k^L as the solution of

$$\mathbf{R}_k z_k = -h_{k-1}.$$

If we use $\mathbf{W}_k = \mathbf{V}_k \mathbf{R}_k^{-1}$ from (19), we have

$$\mathbf{z}_k^L = \mathbf{V}_k z_k^L = -\mathbf{V}_k \mathbf{R}_k^{-1} h_{k-1} = -\mathbf{W}_k h_{k-1} = \mathbf{z}_{k-1}^L - \eta_k \mathbf{w}_k, \tag{32}$$

initialized with $\mathbf{z}_0^L := \mathbf{0}$. By analogy with $\mathbf{y}_{k+1}^C$, we define

$$\bar{\mathbf{W}}_{k+1} := \mathbf{V}_{k+1} \bar{\mathbf{R}}_{k+1}^{-1} = \begin{bmatrix} \mathbf{W}_k & \bar{\mathbf{w}}_{k+1} \end{bmatrix},$$

with $\mathbf{z}_{k+1}^C := \mathbf{V}_{k+1} z_{k+1}^C$, initialized with $\mathbf{z}_1^C := -\bar{\eta}_1 \bar{\mathbf{w}}_1$, and updated according to

$$\mathbf{z}_{k+1}^C = \mathbf{z}_k^L - \bar{\eta}_{k+1} \bar{\mathbf{w}}_{k+1}.$$

The next iteration will update $\mathbf{w}_{k+1} = c_{k+1} \bar{\mathbf{w}}_{k+1}$. Thus z_{k+1}^C solves $\mathbf{T}_{k+1} z = -y_{k+1}^C$, while z_k^L solves

$$\underset{z}{\text{minimize}} \ \|\mathbf{T}_{k+1,k} z + y_k^L\|.$$

In effect, we have approximated the solution of (24) with that of

$$\begin{bmatrix} \mathbf{I}_{k+1} & \mathbf{T}_{k+1,k} \\ \mathbf{T}_{k+1,k}^{\mathsf{T}} & \mathbf{0}_k \end{bmatrix} \begin{bmatrix} y_k \\ z_k \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \gamma_1 \mathbf{e}_1 \end{bmatrix}.$$

As before, $\mathbf{z}_{k+1}^C$ need not be computed at each iteration, but one final step from $\mathbf{z}_k^L$ to $\mathbf{z}_{k+1}^C$ may be taken after convergence has occurred.

3.3. Complete algorithm. The complete procedure, named USYMLQR, is summarized in Algorithm 2. We denote $\mathbf{r}_k$ the residual of (10), which can be updated iteratively or directly as $\mathbf{b} - \mathbf{A}\mathbf{x}_k$ once $\|\mathbf{A}^{\mathsf{T}} \mathbf{r}_k\|$ is sufficiently small. The estimates $\mathbf{y}_k^C$ and $\mathbf{z}_k^C$ need not be updated at each iteration but could be computed if $\|\mathbf{r}_k^C\| < \|\mathbf{r}_{k-1}^L\|$.

Overall, Algorithm 2 stores 6 m-vectors ($\mathbf{u}_k$, $\mathbf{u}_{k+1}$, $\mathbf{q}$, $\mathbf{p}_k$, $\bar{\mathbf{p}}_k$ and $\mathbf{y}_k^L$) if $\mathbf{r}_k$ is not updated iteratively, and 7 n-vectors ($\mathbf{v}_{k-1}$, $\mathbf{v}_k$, $\mathbf{v}_{k+1}$, $\mathbf{w}_{k-1}$, $\mathbf{w}_k$, $\mathbf{x}_k$ and $\mathbf{z}_k^L$). If $\mathbf{y}_{k+1}^C$ should be computed at the end, its storage can be shared with that of $\bar{\mathbf{p}}_{k+1}$. If $\mathbf{z}_{k+1}^C$ should be computed at the end, its storage can be shared with that of $\mathbf{w}_k$ and $\bar{\mathbf{w}}_{k+1}$.

Algorithm 2 USYMLQR: Orthogonal Tridiagonalization for the solution of (4)

Require: $\mathbf{A}$, $\mathbf{b}$, $\mathbf{c}$

Choose stopping tolerances $\varepsilon_{\text{LS}} > 0$ for (10) and $\varepsilon_{\text{LN}} > 0$ for (24)
set `ls_converged` and `ln_converged` to `false`
$\mathbf{u}_0 = \mathbf{0}$, $\mathbf{v}_0 = \mathbf{0}$, $\mathbf{x}_0 = \mathbf{0}$, $\mathbf{w}_0 = \mathbf{0}$
$\beta_1 \mathbf{u}_1 = \mathbf{b}$, and $\gamma_1 \mathbf{v}_1 = \mathbf{c}$ — $(\beta_1, \gamma_1) > 0$ so that $\|\mathbf{u}_1\|_2 = \|\mathbf{v}_1\|_2 = 1$
$\mathbf{q} = \mathbf{A}\mathbf{v}_1$, $\alpha_1 = \mathbf{u}_1^{\mathsf{T}}\mathbf{q}$
$c_0 = -1$, $s_0 = 0$, $\bar{\phi}_1 = \beta_1$, $\lambda_0 = 0$, $\epsilon_{-1} = 0$, $\eta_0 = 0$
$\mathbf{r}_0 = \mathbf{b}$, $\|\mathbf{r}_0\| = \bar{\phi}_1$ — least-squares residual of (10)
$\bar{\delta}_1 = \alpha_1$, $\bar{\mathbf{w}}_1 = \mathbf{v}_1/\bar{\delta}_1$, $\mathbf{z}_0^L = \mathbf{0}$, $\mathbf{z}_1^C = -\bar{\eta}_1 \bar{\mathbf{w}}_1$
$\bar{\eta}_1 = \gamma_1/\bar{\delta}_1$, $\bar{\mathbf{p}}_1 = \mathbf{u}_1$, $\mathbf{y}_0^L = \mathbf{0}$, $\mathbf{y}_1^C = \bar{\eta}_1 \bar{\mathbf{p}}_1$
for $k = 1, 2, \ldots$ **do**
  $\beta_{k+1}\mathbf{u}_{k+1} = \mathbf{q} - \alpha_k \mathbf{u}_k$ — $\beta_{k+1} > 0$ so that $\|\mathbf{u}_{k+1}\|_2 = 1$
  $\gamma_{k+1}\mathbf{v}_{k+1} = \mathbf{A}^{\mathsf{T}}\mathbf{u}_k - \beta_k \mathbf{v}_{k-1} - \alpha_k \mathbf{v}_k$ — $\gamma_{k+1} > 0$ so that $\|\mathbf{v}_{k+1}\|_2 = 1$
  $\bar{\lambda}_k = -c_{k-1}\gamma_{k+1}$, $\epsilon_{k-1} = s_{k-1}\gamma_{k+1}$ — continue QR factorization
  $\delta_k = (\bar{\delta}_k^2 + \beta_{k+1}^2)^{\frac{1}{2}}$, $c_k = \bar{\delta}_k/\delta_k$, $s_k = \beta_{k+1}/\delta_k$
  $\mathbf{w}_k = c_k \bar{\mathbf{w}}_k$
  if `ls_converged` is `false` **then**
    $\|\mathbf{A}^{\mathsf{T}}\mathbf{r}_{k-1}\| = |\bar{\phi}_k|\sqrt{\bar{\delta}_k^2 + \bar{\lambda}_k^2}$ — optimality residual of (10) at $\mathbf{x}_{k-1}$
    `ls_converged` $= \|\mathbf{A}^{\mathsf{T}}\mathbf{r}_{k-1}\| \leqslant \varepsilon_{\text{LS}}$
  end if
  if `ls_converged` is `false` **then**
    $\phi_k = c_k \bar{\phi}_k$, $\bar{\phi}_{k+1} = s_k \bar{\phi}_k$
    $\mathbf{x}_k = \mathbf{x}_{k-1} + \phi_k \mathbf{w}_k$ — update solution of (10)
    $\mathbf{r}_k = s_k^2 \mathbf{r}_{k-1} - c_k \bar{\phi}_k \mathbf{u}_{k+1}$, $\|\mathbf{r}_k\| = |\bar{\phi}_{k+1}|$ — residual of (10) at $\mathbf{x}_k$
  end if
  $\mathbf{q} = \mathbf{A}\mathbf{v}_{k+1} - \gamma_{k+1}\mathbf{u}_k$, $\alpha_{k+1} = \mathbf{u}_{k+1}^{\mathsf{T}}\mathbf{q}$
  $\lambda_k = c_k \bar{\lambda}_k + s_k \alpha_{k+1}$, $\bar{\delta}_{k+1} = s_k \bar{\lambda}_k - c_k \alpha_{k+1}$
  if `ln_converged` is `false` **then**
    $\eta_k = c_k \bar{\eta}_k$
    $\|\mathbf{r}_{k-1}^L\| = \left((\delta_k \eta_k)^2 + (\epsilon_{k-1}\eta_{k-1})^2\right)^{\frac{1}{2}}$ — residual of (24) at $\mathbf{y}_{k-1}^L$
    `ln_converged` $= \|\mathbf{r}_{k-1}^L\| \leqslant \varepsilon_{\text{LN}}$
  end if
  $\bar{\mathbf{w}}_{k+1} = (\mathbf{v}_{k+1} - \lambda_k \mathbf{w}_k - \epsilon_{k-1}\mathbf{w}_{k-1})/\bar{\delta}_{k+1}$
  if `ln_converged` is `false` **then**
    $\mathbf{p}_k = c_k \bar{\mathbf{p}}_k + s_k \mathbf{u}_{k+1}$
    $\mathbf{y}_k^L = \mathbf{y}_{k-1}^L + \eta_k \mathbf{p}_k$ — update LQ solution of (24)
    $\mathbf{z}_k^L = \mathbf{z}_{k-1}^L - \eta_k \mathbf{w}_k$ — update LQ multipliers of (24)
    $\|\mathbf{r}_k^C\| = \gamma_{k+1}|s_{k-1}\eta_{k-1} - c_{k-1}\bar{\eta}_k|$ — residual of (24) at $\mathbf{y}_k^C$
    $\bar{\mathbf{p}}_{k+1} = s_k \bar{\mathbf{p}}_k - c_k \mathbf{u}_{k+1}$
    $\bar{\eta}_{k+1} = -(\lambda_k \eta_k + \epsilon_{k-1}\eta_{k-1})/\bar{\delta}_{k+1}$
    $\mathbf{y}_{k+1}^C = \mathbf{y}_k^L + \bar{\eta}_{k+1}\bar{\mathbf{p}}_{k+1}$ — update CG solution of (24)
    $\mathbf{z}_{k+1}^C = \mathbf{z}_k^L - \bar{\eta}_{k+1}\bar{\mathbf{w}}_{k+1}$ — update CG multipliers of (24)
  end if
end for
return $(\mathbf{s}_k, \mathbf{t}_k) = (\mathbf{r}_k, \mathbf{x}_k) + (\mathbf{y}_k^L, \mathbf{z}_k^L)$ (or $(\mathbf{s}_k, \mathbf{t}_k) = (\mathbf{r}_k, \mathbf{x}_k) + (\mathbf{y}_k^C, \mathbf{z}_k^C)$)

TABLE 1
Storage and work per iteration of Algorithm 2 and MINRES *for the solution of* (4).

	vectors	dots/iter	scal/iter	axpy/iter
USYMLQR	$6m + 7n$	$1m$	$2m$	$5m + 4n$
MINRES	$7(m + n)$	$1(m + n)$	$1(n + m)$	$6(m + n)$

For comparison, MINRES requires 7 $(n + m)$-vectors of storage, which amounts to one extra m-vector and we are assuming that $m \geqslant n$. A tally of storage and work for Algorithm 2 and MINRES appears in Table 1. In Table 1, "dots/iter" refers to the number of dot products per iterations, "scal/iter" refers to the number of operations of the form $x \leftarrow \alpha x$ per iteration, where x is a vector and α a scalar, beyond normalization of the basis vectors, and "axpy/iter" refers to the number of operations of the form $x \leftarrow x + \alpha y$ per iteration, where y is a vector. The factors of m and n indicate the number of such operations on m-vectors and n-vectors, respectively. Table 1 shows that the storage and work per iteration is comparable to, or slightly lower than, MINRES.

4. Estimation of norms.

4.1. Computing $\|\mathbf{x}\|$. An estimate of $\|\mathbf{x}_k\|$ may be obtained as in (Paige and Saunders, 1982, §5.2). It is possible to reduce $\mathbf{R}_k$ to lower triangular form using appropriate reflections, i.e., $\mathbf{R}_k \widetilde{\mathbf{Q}}_k^{\mathsf{T}} = \widetilde{\mathbf{L}}_k$, where $\widetilde{\mathbf{L}}_k$ is lower triangular with three diagonals. Define $\widetilde{p}_k$ as the solution of $\widetilde{\mathbf{L}}_k \widetilde{p}_k = f_k$ and note that

$$\mathbf{x}_k = \mathbf{V}_k x_k = \mathbf{V}_k \mathbf{R}_k^{-1} f_k = \mathbf{V}_k \widetilde{\mathbf{Q}}_k^{\mathsf{T}} \widetilde{\mathbf{L}}_k^{-1} f_k = \mathbf{V}_k \widetilde{\mathbf{Q}}_k^{\mathsf{T}} \widetilde{p}_k.$$

By orthogonality, $\|\mathbf{x}_k\| = \|\widetilde{p}_k\|$, which is easily accumulated. If we denote

$$\widetilde{\mathbf{L}}_k = \begin{bmatrix} \dot{\delta}_1 & & & & \\ \dot{\lambda}_1 & \dot{\delta}_2 & & & \\ \dot{\epsilon}_1 & \dot{\lambda}_2 & \dot{\delta}_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \dot{\epsilon}_{k-2} & \dot{\lambda}_{k-1} & \widetilde{\delta}_k \end{bmatrix},$$

we find $\widetilde{p}_k = (\pi_1, \ldots, \pi_{k-1}, \widetilde{\pi}_k)$ recursively as

$$\pi_1 = \frac{\phi_1}{\dot{\delta}_1}, \quad \pi_2 = \frac{\phi_2 - \dot{\lambda}_1 \pi_1}{\dot{\delta}_2}, \quad \pi_j = \frac{\phi_j - \dot{\lambda}_{j-1} \pi_{j-1} - \dot{\epsilon}_{j-2} \pi_{j-2}}{\dot{\delta}_j}, \ (j = 3, \ldots, k-1),$$

and

$$\widetilde{\pi}_k = \frac{\phi_k - \dot{\lambda}_{k-1} \pi_{k-1} - \dot{\epsilon}_{k-2} \pi_{k-2}}{\widetilde{\delta}_k}.$$

Thus, we may update an accumulator $\xi_{k-1}^2 := \pi_1^2 + \cdots + \pi_{k-1}^2$ and

$$\|\mathbf{x}_k\|^2 = \xi_{k-1}^2 + \widetilde{\pi}_k^2. \tag{33}$$

Two additional reflections per iteration are required to reduce $\mathbf{R}_k$ to lower triangular form. The first reflection, designed to zero out the first superdiagonal, can be

represented as

$$\begin{array}{c} \\ k \\ k+1 \end{array}\!\!\begin{array}{c} \begin{array}{ccc} k & k+1 & k+2 \end{array} \\ \begin{bmatrix} \breve{\delta}_k & \breve{\lambda}_k & \epsilon_k \\ & \breve{\delta}_{k+1} & \lambda_{k+1} \end{bmatrix} \end{array} \begin{array}{c} \begin{array}{cc} k & k+1 \end{array} \\ \begin{bmatrix} \breve{c}_k & \breve{s}_k \\ \breve{s}_k & -\breve{c}_k \end{bmatrix} \end{array} = \begin{array}{c} \begin{array}{ccc} k & k+1 & k+2 \end{array} \\ \begin{bmatrix} \ddot{\delta}_k & 0 & \epsilon_k \\ \ddot{\lambda}_k & \breve{\delta}_{k+1} & \lambda_{k+1} \end{bmatrix} \end{array},$$

and is defined by

$$\ddot{\delta}_k = \sqrt{\breve{\delta}_k^2 + \breve{\lambda}_k^2}, \quad \breve{c}_k = \breve{\delta}_k / \ddot{\delta}_k, \quad \breve{s}_k = \breve{\lambda}_k^2 / \ddot{\delta}_k.$$

The first reflection produces

$$\ddot{\lambda}_k = \breve{s}_k \breve{\delta}_{k+1}, \quad \text{and} \quad \breve{\delta}_{k+1} = -\breve{c}_k \breve{\delta}_{k+1},$$

and is initialized with $\breve{\delta}_1 = \delta_1$, $\breve{\lambda}_1 = \lambda_1$, and $\breve{\delta}_2 = \delta_2$.

The second reflection, designed to zero out the second superdiagonal, can be represented as

$$\begin{array}{c} \\ k \\ k+1 \\ k+2 \end{array}\!\!\begin{array}{c} \begin{array}{ccc} k & k+1 & k+2 \end{array} \\ \begin{bmatrix} \ddot{\delta}_k & 0 & \epsilon_k \\ \ddot{\lambda}_k & \breve{\delta}_{k+1} & \lambda_{k+1} \\ & & \delta_{k+2} \end{bmatrix} \end{array} \begin{array}{c} \begin{array}{cc} k & k+2 \end{array} \\ \begin{bmatrix} \dot{c}_k & \dot{s}_k \\ \dot{s}_k & -\dot{c}_k \end{bmatrix} \end{array} = \begin{array}{c} \begin{array}{ccc} k & k+1 & k+2 \end{array} \\ \begin{bmatrix} \dot{\delta}_k & 0 & 0 \\ \dot{\lambda}_k & \breve{\delta}_{k+1} & \breve{\lambda}_{k+1} \\ \dot{\epsilon}_k & & \breve{\delta}_{k+2} \end{bmatrix} \end{array},$$

and is defined by

$$\dot{\delta}_k = \sqrt{\ddot{\delta}_k^2 + \epsilon_k^2}, \quad \dot{c}_k = \ddot{\delta}_k / \dot{\delta}_k, \quad \dot{s}_k = \epsilon_k / \dot{\delta}_k.$$

It produces

$$\dot{\lambda}_k = \dot{c}_k \ddot{\lambda}_k + \dot{s}_k \lambda_{k+1}, \quad \dot{\epsilon}_k = \dot{s}_k \delta_{k+2}, \quad \breve{\lambda}_{k+1} = \dot{s}_k \ddot{\lambda}_k - \dot{c}_k \lambda_{k+1}, \quad \breve{\delta}_{k+2} = -\dot{c}_k \delta_{k+2},$$

and thus the k-th column of $\mathbf{L}_k$.

4.2. Estimating $\|\mathbf{y}\|$. Because both $\mathbf{y}_k^L$ and $\mathbf{y}_k^C$ are updated using orthonormal directions, we have

$$\|\mathbf{y}_k^L\|^2 = \sum_{j=1}^{k-1} \eta_j^2 \qquad \text{and} \qquad \|\mathbf{y}_{k+1}^C\|^2 = \sum_{j=1}^{k-1} \eta_j^2 + \bar{\eta}_k^2.$$

In exact arithmetic, the minimum-norm solution $\mathbf{y}_*$ will be identified after at most n, iterations, i.e., $\mathbf{y}_* = \mathbf{y}_{n+1}^L$ so that the error $\mathbf{e}_k^L := \mathbf{y}_* - \mathbf{y}_k^L$ satisfies

$$\|\mathbf{e}_k^L\|^2 = \sum_{j=k}^{n} \eta_j^2.$$

If monitoring the error is of interest, Hestenes and Stiefel (1952) suggest choosing a small delay $d \in \mathbb{N}_0$ and a tolerance $\varepsilon > 0$, and using the stopping condition

$$\sum_{j=k-d+1}^{k} \eta_j^2 \leqslant \varepsilon^2 \sum_{j=1}^{k} \eta_j^2.$$

The left-hand side of the stopping test yields a lower bound on $\|\mathbf{e}_{k-d+1}^L\|^2$.

4.3. Estimating $\|\mathbf{A}\|$ and cond$(\mathbf{A})$. We have from (5a) that

$$\mathbf{V}_k^T \mathbf{A}^T \mathbf{A} \mathbf{V}_k = \mathbf{T}_{k+1,k}^T \mathbf{T}_{k+1,k},$$

so that the eigenvalues of $\mathbf{T}_{k+1,k}^T \mathbf{T}_{k+1,k}$ interlace those of $\mathbf{A}^T\mathbf{A}$. Consequently, the singular values of $\mathbf{T}_{k+1,k}$ also interlace those of $\mathbf{A}$, and we have $\|\mathbf{T}_{k+1,k}\|_F \leqslant \|\mathbf{A}\|_F$. It is easy to accumulate $\|\mathbf{T}_{k+1,k}\|_F$ during the iterations of Algorithm 1 using the recursion

$$\|\mathbf{T}_{k+1,k}\|_F^2 = \|\mathbf{T}_{k,k-1}\|_F^2 + \gamma_k^2 + \alpha_k^2 + \beta_{k+1}^2.$$

In USYMQR, we may derive an estimate of cond$(\mathbf{A})$ as in (Paige and Saunders, 1982). The factorization (11) yields $\mathbf{T}_{k+1,k}^T \mathbf{T}_{k+1,k} = \mathbf{R}_k^T \mathbf{R}_k$, and therefore

$$\|\mathbf{R}_k^{-1}\|_F = \|\mathbf{T}_{k+1,k}^+\|_F \leqslant \|\mathbf{A}^+\|_F.$$

Using now (19), $\|\mathbf{T}_{k+1,k}^+\|_F = \|\mathbf{R}_k^{-1}\|_F = \|\mathbf{W}_k\|_F$ and

$$\text{cond}(\mathbf{T}_{k+1,k}) = \|\mathbf{T}_{k+1,k}\|_F \|\mathbf{T}_{k+1,k}^+\|_F = \|\mathbf{T}_{k+1,k}\|_F \|\mathbf{W}_k\|_F \leqslant \text{cond}(\mathbf{A}).$$

5. Backward error analysis. One way to determine whether a *computed* solution $(\mathbf{s}, \mathbf{t})$ is a good enough approximate solution to (4) might be to consider the residual norm. It is well known that the residual norm is related to the normwise *backward error*: if

$$\left\| \begin{bmatrix} \mathbf{b} \\ \mathbf{c} \end{bmatrix} - \begin{bmatrix} \mathbf{I} & \mathbf{A} \\ \mathbf{A}^T & \end{bmatrix} \begin{bmatrix} \mathbf{s} \\ \mathbf{t} \end{bmatrix} \right\| \leqslant \alpha \left\| \begin{bmatrix} \mathbf{I} & \mathbf{A} \\ \mathbf{A}^T & \end{bmatrix} \right\|_F \left\| \begin{bmatrix} \mathbf{s} \\ \mathbf{t} \end{bmatrix} \right\| + \beta \left\| \begin{bmatrix} \mathbf{b} \\ \mathbf{c} \end{bmatrix} \right\|$$

then there exist perturbations $\Delta\boldsymbol{\mathcal{A}}$ and $\Delta\boldsymbol{\mathcal{B}}$ such that

$$\left(\begin{bmatrix} \mathbf{I} & \mathbf{A} \\ \mathbf{A}^T & \end{bmatrix} + \Delta\boldsymbol{\mathcal{A}} \right) \begin{bmatrix} \mathbf{s} \\ \mathbf{t} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{c} \end{bmatrix} + \Delta\boldsymbol{\mathcal{B}}, \tag{34}$$

with

$$\|\Delta\boldsymbol{\mathcal{A}}\|_F \leqslant \alpha \left\| \begin{bmatrix} \mathbf{I} & \mathbf{A} \\ \mathbf{A}^T & \end{bmatrix} \right\|_F, \qquad \|\Delta\boldsymbol{\mathcal{B}}\| \leqslant \beta \left\| \begin{bmatrix} \mathbf{b} \\ \mathbf{c} \end{bmatrix} \right\|,$$

(Rigal and Gaches, 1967). However, the perturbation $\Delta\boldsymbol{\mathcal{A}}$ in (34) does not necessarily have the same block structure as the original matrix. As we are solving a structured problem using a structured approach, we believe it is more appropriate to consider the *structured* backward error. We seek perturbations in the data of (4) that maintain the saddle-point structure, i.e., perturbations of the form

$$\begin{bmatrix} \mathbf{I} & \mathbf{A} + \Delta\mathbf{A} \\ \mathbf{A}^T + \Delta\mathbf{A}^T & \end{bmatrix} \begin{bmatrix} \mathbf{s} \\ \mathbf{t} \end{bmatrix} = \begin{bmatrix} \mathbf{b} + \Delta\mathbf{b} \\ \mathbf{c} + \Delta\mathbf{c} \end{bmatrix}. \tag{35}$$

Given a *computed* solution $(\mathbf{s}, \mathbf{t})$ of (4), a structured backward error analysis asks the question: is $(\mathbf{s}, \mathbf{t})$ the *exact* solution of a *nearby* system of the form (35)? If the resulting perturbations $\Delta\mathbf{A}$, $\Delta\mathbf{b}$, and $\Delta\mathbf{c}$ can be made small enough relative to $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{c}$, then we may be satisfied with $(\mathbf{s}, \mathbf{t})$ as a computed solution of (4).

Clearly, the condition (35) is more stringent than (34). Sun (1999) gives examples in which the structured backward error for saddle point problems is arbitrarily larger than the unstructured one. Extending the results of Sun (1999), Xiang and Wei (2007)

define a structured nearness measure $\gamma_{\lambda,\mu}(\mathbf{s},\mathbf{t})$ as the optimal objective value of the constrained optimization problem

$$\gamma_{\lambda,\mu}(\mathbf{s},\mathbf{t}) := \begin{cases} \underset{\Delta\mathbf{A},\Delta\mathbf{b},\Delta\mathbf{c}}{\text{minimize}} & \left(\|\Delta\mathbf{A}\|_F^2 + \lambda^2\|\Delta\mathbf{b}\|^2 + \mu^2\|\Delta\mathbf{c}\|^2\right)^{\frac{1}{2}} \\ \text{subject to} & \begin{bmatrix} \mathbf{I} & \mathbf{A}+\Delta\mathbf{A} \\ \mathbf{A}^\mathsf{T}+\Delta\mathbf{A}^\mathsf{T} & \end{bmatrix} \begin{bmatrix} \mathbf{s} \\ \mathbf{t} \end{bmatrix} = \begin{bmatrix} \mathbf{b}+\Delta\mathbf{b} \\ \mathbf{c}+\Delta\mathbf{c} \end{bmatrix}, \end{cases} \tag{36}$$

where λ and μ are weights that may be adjusted to emphasize one perturbation more than another. An interesting selection is $\lambda_\star := \|\mathbf{A}\|_F/\|\mathbf{b}\|$ and $\mu_\star := \|\mathbf{A}\|_F/\|\mathbf{c}\|$, which yields the normwise relative measure

$$\gamma_S(\mathbf{s},\mathbf{t}) = \left(\left(\frac{\|\Delta\mathbf{A}\|_F}{\|\mathbf{A}\|_F}\right)^2 + \left(\frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|}\right)^2 + \left(\frac{\|\Delta\mathbf{c}\|}{\|\mathbf{c}\|}\right)^2\right)^{\frac{1}{2}} = \|\mathbf{A}\|_F^{-1}\,\gamma_{\lambda_\star,\mu_\star}(\mathbf{s},\mathbf{t}). \tag{37}$$

If $\gamma_S(\mathbf{s},\mathbf{t})$ is smaller than a chosen tolerance, then $(\mathbf{s},\mathbf{t})$ is the exact solution of a nearby system of the form (35) to within the same tolerance. This criterion can be used to determine when to stop the iteration.

Xiang and Wei (2007) establish that

$$\begin{aligned} \gamma_{\lambda,\mu}(\mathbf{s},\mathbf{t})^2 = {} & \frac{\lambda^2}{\theta_\lambda}\|\mathbf{r}_b\|^2 + \frac{\mu^2}{\theta_\mu}\|\mathbf{r}_c\|^2 - 2\frac{\lambda^2\mu^2}{\theta}(\mathbf{r}_b^\mathsf{T}\mathbf{s})(\mathbf{r}_c^\mathsf{T}\mathbf{t}) \\ & + \frac{\lambda^2\mu^2(\theta_\lambda-1)}{\theta_\lambda\theta}(\mathbf{r}_b^\mathsf{T}\mathbf{s})^2 + \frac{\lambda^2\mu^2(\theta_\mu-1)}{\theta_\mu\theta}(\mathbf{r}_c^\mathsf{T}\mathbf{t})^2, \end{aligned} \tag{38}$$

where $\mathbf{r}_b := \mathbf{b}-\mathbf{s}-\mathbf{A}\mathbf{t}$, $\mathbf{r}_c := \mathbf{c}-\mathbf{A}^\mathsf{T}\mathbf{s}$, $\theta_\lambda := 1+\lambda^2\|\mathbf{t}\|^2$, $\theta_\mu := 1+\mu^2\|\mathbf{s}\|^2$, and $\theta := 1+\lambda^2\|\mathbf{t}\|^2+\mu^2\|\mathbf{s}\|^2$.

It is also possible to monitor the convergence of the least-squares subproblem (10) and the least-norm subproblem (24) separately. This approach simplifies the stopping criterion, and can be justified as follows. Suppose $(\mathbf{r},\mathbf{x})$ and $(\mathbf{y},\mathbf{z})$ are good approximate solutions to (10) and (24), respectively, in the sense that

$$\begin{bmatrix} \mathbf{I} & \mathbf{A}+\Delta\mathbf{A}_1 \\ \mathbf{A}^\mathsf{T}+\Delta\mathbf{A}_1^\mathsf{T} & \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{b}+\Delta\mathbf{b} \\ \mathbf{0} \end{bmatrix}, \quad \text{and}$$

$$\begin{bmatrix} \mathbf{I} & \mathbf{A}+\Delta\mathbf{A}_2 \\ \mathbf{A}^\mathsf{T}+\Delta\mathbf{A}_2^\mathsf{T} & \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{c}+\Delta\mathbf{c} \end{bmatrix},$$

where

$$\frac{\|\Delta\mathbf{A}_1\|_F}{\|\mathbf{A}\|_F}, \frac{\|\Delta\mathbf{A}_2\|_F}{\|\mathbf{A}\|_F}, \frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|}, \frac{\|\Delta\mathbf{c}\|}{\|\mathbf{c}\|} \leqslant \epsilon.$$

Let $(\mathbf{s},\mathbf{t}) = (\mathbf{r},\mathbf{x})+(\mathbf{y},\mathbf{z})$. It is straightforward to verify that there exist perturbations $\Delta\mathbf{b}_2$ and $\Delta\mathbf{c}_2$ such that

$$\begin{bmatrix} \mathbf{I} & \mathbf{A}+\Delta\mathbf{A}_2 \\ \mathbf{A}^\mathsf{T}+\Delta\mathbf{A}_2^\mathsf{T} & \end{bmatrix} \begin{bmatrix} \mathbf{s} \\ \mathbf{t} \end{bmatrix} = \begin{bmatrix} \mathbf{b}+\Delta\mathbf{b}_2 \\ \mathbf{c}+\Delta\mathbf{c}_2 \end{bmatrix},$$

where

$$\|\Delta\mathbf{b}_2\| \leqslant \epsilon(\|\mathbf{b}\|+2\|\mathbf{A}\|_F\|\mathbf{x}\|), \quad \|\Delta\mathbf{c}_2\| \leqslant \epsilon(\|\mathbf{c}\|+2\|\mathbf{A}\|_F\|\mathbf{r}\|).$$

In other words, provided the problem is not too badly scaled (in that $\|\mathbf{A}\|_F\|\mathbf{x}\|$ is not much larger than $\|\mathbf{b}\|$ and $\|\mathbf{A}\|_F\|\mathbf{r}\|$ is not much larger than $\|\mathbf{c}\|$), then $(\mathbf{s},\mathbf{t})$ is a good approximate solution as per (35).

We provide details on the two separate subproblems in the next sections.

5.1. Least-Squares Subproblem. In the least-squares problem, $\mathbf{c} = \mathbf{0}$ and we impose $\Delta\mathbf{c} = \mathbf{0}$. In the notation of (10), we rename $\mathbf{s} \leftarrow \mathbf{r}$ and $\mathbf{t} \leftarrow \mathbf{x}$ in (35). In other words, we seek peturbations of the form

$$\begin{bmatrix} \mathbf{I} & \mathbf{A} + \Delta\mathbf{A} \\ \mathbf{A}^T + \Delta\mathbf{A}^T & \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{b} + \Delta\mathbf{b} \\ \mathbf{0} \end{bmatrix}.$$

Xiang and Wei (2007) indicate that the relevant measure results from taking the limit when $\mu \to \infty$ in (38). In addition, in USYMQR, $\mathbf{r}_b = \mathbf{b} - \mathbf{r} - \mathbf{A}\mathbf{x} = \mathbf{0}$ by construction—see (20). Thus,

$$\gamma_{\lambda_\star,\infty}(\mathbf{r},\mathbf{x})^2 = \frac{\|\mathbf{A}^T\mathbf{r}\|^2}{\|\mathbf{r}\|^2} + \frac{\|\mathbf{A}\|_F^2}{\|\mathbf{b}\|^2\,\|\mathbf{r}\|^2}(\mathbf{x}^T\mathbf{A}^T\mathbf{r})^2$$

and, as in (37),

$$\gamma_{\text{LS}}(\mathbf{r},\mathbf{x}) := \|\mathbf{A}\|_F^{-1}\,\gamma_{\lambda_\star,\infty}(\mathbf{r},\mathbf{x}) = \left(\frac{\|\mathbf{A}^T\mathbf{r}\|^2}{\|\mathbf{A}\|_F^2\,\|\mathbf{r}\|^2} + \frac{(\mathbf{x}^T\mathbf{A}^T\mathbf{r})^2}{\|\mathbf{b}\|^2\,\|\mathbf{r}\|^2}\right)^{\frac{1}{2}}. \tag{39}$$

In theory, in USYMQR, $\mathbf{x}_k \in \text{Range}(\mathbf{V}_k)$ while $\mathbf{A}^T\mathbf{r}_k$ is a combination of $\mathbf{v}_{k+1}$ and $\mathbf{v}_{k+2}$, and thus $\mathbf{x}_k^T\mathbf{A}^T\mathbf{r}_k = \mathbf{0}$. Unfortunately, in finite-precision arithmetic, orthogonality is soon lost, and the second term in (39) may contribute to the backward error. In any case, note that

$$\frac{\|\mathbf{A}^T\mathbf{r}\|}{\|\mathbf{A}\|_F\,\|\mathbf{r}\|} \leqslant \gamma_{\text{LS}}(\mathbf{r},\mathbf{x}) \leqslant \frac{\|\mathbf{A}^T\mathbf{r}\|}{\|\mathbf{A}\|_F\,\|\mathbf{r}\|}\left(1 + \frac{\|\mathbf{A}\|_F^2\|\mathbf{x}\|^2}{\|\mathbf{b}\|^2}\right)^{\frac{1}{2}}.$$

Thus, provided $\|\mathbf{A}\|_F\|\mathbf{x}\|$ is not much larger than $\|\mathbf{b}\|$, we can accept $(\mathbf{r},\mathbf{x})$ as computed solution and stop updating $(\mathbf{r},\mathbf{x})$ when

$$\frac{\|\mathbf{A}^T\mathbf{r}\|}{\|\mathbf{A}\|_F\,\|\mathbf{r}\|} \leqslant \text{tol}. \tag{40}$$

This stopping condition is often used in the iterative solution of least-squares problems (Paige and Saunders, 1982; Fong and Saunders, 2011). If $\|\mathbf{A}\|_F\|\mathbf{x}\| \gg \|\mathbf{b}\|$, the backward error (39) can be computed exactly at the cost of an extra dot product between $\mathbf{x}$ and $\mathbf{A}^T\mathbf{r}$ as given in (22).

5.2. Least-Norm Subproblem. In the least-norm problem, we have $\mathbf{b} = \mathbf{0}$ and impose $\Delta\mathbf{b} = \mathbf{0}$. In the notation of (24), we rename $\mathbf{s} \leftarrow \mathbf{y}$ and $\mathbf{t} \leftarrow \mathbf{z}$ in (35). In other words, we seek perturbations of the form

$$\begin{bmatrix} \mathbf{I} & \mathbf{A} + \Delta\mathbf{A} \\ \mathbf{A}^T + \Delta\mathbf{A}^T & \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{c} + \Delta\mathbf{c} \end{bmatrix}.$$

Additionally, $\mathbf{r}_b = -(\mathbf{y} + \mathbf{A}\mathbf{z}) = \mathbf{0}$ by construction of $\mathbf{z}$—see subsection 3.2.3. We take the limit as $\lambda \to \infty$ in (36) and obtain

$$\gamma_{\infty,\mu_\star}(\mathbf{y},\mathbf{z})^2 = \frac{\|\mathbf{A}\|_F^2\,\|\mathbf{c} - \mathbf{A}^T\mathbf{y}\|^2}{\|\mathbf{c}\|^2 + \|\mathbf{A}\|_F^2\|\mathbf{y}\|^2} + \frac{\|\mathbf{A}\|_F^4\,\|\mathbf{y}\|^2}{\|\mathbf{c}\|^2(\|\mathbf{c}\|^2 + \|\mathbf{A}\|_F^2\|\mathbf{y}\|^2)}\left((\mathbf{c} - \mathbf{A}^T\mathbf{y})^T\mathbf{z}\right)^2,$$

and, as in (37) and (39),

$$\gamma_{\text{LN}}(\mathbf{y}, \mathbf{z}) = \|\mathbf{A}\|_F^{-1} \, \gamma_{\infty,\mu_\star}(\mathbf{y}, \mathbf{z})$$

$$= \left(\frac{\|\mathbf{c} - \mathbf{A}^{\mathsf{T}}\mathbf{y}\|^2}{\|\mathbf{c}\|^2 + \|\mathbf{A}\|_F^2 \, \|\mathbf{y}\|^2} + \frac{\|\mathbf{A}\|_F^2 \, \|\mathbf{y}\|^2}{\|\mathbf{c}\|^2 (\|\mathbf{c}\|^2 + \|\mathbf{A}\|_F^2 \, \|\mathbf{y}\|^2)} \left((\mathbf{c} - \mathbf{A}^{\mathsf{T}}\mathbf{y})^{\mathsf{T}} \mathbf{z} \right)^2 \right)^{\frac{1}{2}}. \tag{41}$$

By construction, $\mathbf{z}_k \in \text{Range}(\mathbf{V}_k)$ while $\mathbf{c} - \mathbf{A}^{\mathsf{T}}\mathbf{y}$ is a combination of $\mathbf{v}_{k+1}$ and $\mathbf{v}_{k+2}$. If orthogonality is maintained, the above expression reduces to

$$\gamma_{\text{LN}}(\mathbf{y}, \mathbf{z}) = \frac{\|\mathbf{c} - \mathbf{A}^{\mathsf{T}}\mathbf{y}\|}{\sqrt{\|\mathbf{c}\|^2 + \|\mathbf{A}\|_F^2 \, \|\mathbf{y}\|^2}}, \tag{42}$$

which is similar to the unstructured normwise relative backward error for $\mathbf{A}^{\mathsf{T}}\mathbf{y} = \mathbf{c}$. It is also possible to implement (41) directly at the expense of an extra dot product between (29) and (32), or to bound $(\mathbf{c} - \mathbf{A}^{\mathsf{T}}\mathbf{y})^{\mathsf{T}}\mathbf{z} \leqslant \|\mathbf{c} - \mathbf{A}^{\mathsf{T}}\mathbf{y}\| \, \|\mathbf{z}\|$.

6. Numerical experiments. Our implementation of USYMLQR in the Julia[1] language is available from `github.com/JuliaSmoothOptimizers/Krylov.jl`. We stop updating $\mathbf{x}$ as soon as convergence occurs for (10), and stop updating $\mathbf{y}$ and $\mathbf{z}$ as soon as convergence occurs for (24). When one of the two subproblems is solved, subsequent iterations only generate the quantities necessary to update the iterates of the other one. We compute the residual (20) once at the end instead of updating it along the iterations.

We perform preliminary experiments on rectangular matrices from the SuiteSparse Matrix Collection[2] (Davis and Hu, 2011) that provide an accompanying right-hand side and compare our results with MINRES . When a tall and skinny matrix is read, it plays the role of $\mathbf{A}$ while a short and wide matrix plays the role of $\mathbf{A}^{\mathsf{T}}$. In each case, we scale the matrix so the columns of the tall matrix have unit norm. For each matrix, we construct (4) where $\mathbf{b}$ and $\mathbf{c}$ are the accompanying right-hand side and the vectors of ones. The overall right-hand side $(\mathbf{b}, \mathbf{c})$ is subsequently normalized.

For MINRES applied to consistent systems, we use as convergence criterion

$$\frac{\|\bar{\mathbf{r}}_k\|}{\|\mathbf{K}\| \, \|(\mathbf{s}_k, \mathbf{t}_k)\|} \leqslant \epsilon, \tag{43}$$

where $\mathbf{K}$ and $\bar{\mathbf{r}}$ are the matrix and the residual of (4), respectively, $\|\mathbf{K}\|$ is approximated by the running estimate of the norm of $\mathbf{K}$, and $\epsilon > 0$ is a user-chosen tolerance. The first condition is the optimality condition corresponding to a minimum-norm residual while the second applies to zero-residual problems. Likewise, USYMLQR uses the stopping condition $\gamma_{\text{LS}} \leqslant \epsilon$ and $\gamma_{\text{LN}} \leqslant \epsilon$, where γ_{LS} and γ_{LN} are defined in (40) and (42), respectively. USYMLQR is also equipped with a stopping condition for zero-residual problems similar to that of MINRES but it was never triggered in the experiments below. All our experiments use $\epsilon = 10^{-8}$. In the case of MINRES, this corresponds to setting `atol=1.0e-8` and `rtol=0`. Because the subspaces explored by USYMLQR are related to those explored by methods based on the Golub and Kahan (1965) process, we include convergence curves corresponding to LSQR (Paige and Saunders, 1982) and the method of Craig (1955) for comparison purposes. The reader should keep in mind

[1] `julialang.org`

[2] Formerly the University of Florida Sparse Matrix Collection.

that LSQR and CRAIG each solve one of (3), while USYMLQR solves both simultaneously. The maximum number of iterations of USYMLQR, LSQR and CRAIG is set to the larger dimension of $\mathbf{A}$ while the maximum number of MINRES iterations is set to $m + n$.

The figures report the backward error appropriate for each method: for LSQR and the least-squares part of USYMLQR, we report (40), for CRAIG and the least-norm part of USYMLQR, we report (42), and for MINRES, we report (43).

Figure 1 and Figure 2 summarize the results for two over-determined problems arising from a least-squares application and two under-determined problems arising from linear optimization. The figures make it apparent that USYMLQR stops updating the solution of one of (10) and (24) before the other. On problem wellc1850, USYMQR and USYMLQ terminate after 456 and 495 iterations, respectively, while MINRES terminates after 699 iterations. On problem illc1850, those numbers are $1,204$, $1,647$ and $2,199$, respectively. The situation is similar for the remaining problems.

Problem lp_d6cube is row rank deficient but the under-determined system is nonetheless consistent and convergence occurs in a small number of iterations. Problem lp_czprob has full row rank and is consistent. In both cases, we observe that MINRES requires more iterations to converge when the stopping tolerance is tight. If a loose tolerance is appropriate, MINRES might terminate earlier than USYMLQR.

It is clear in the plots that neither $\|\mathbf{A}^T\mathbf{r}\|$ nor $\|\mathbf{A}^T\mathbf{y} - \mathbf{c}\|$ is monotonic, while the MINRES residual is monotonic by design. However, the results illustrate the fact that USYMLQR may terminate in fewer iterations, and therefore fewer operator-vector products, than MINRES. Although certain curves show a staircase behavior, it is not clear that there is a relation between the MINRES iterations and those of USYMLQR.

We caution the reader that we explicitly assume that (4) is consistent. On inconsistent systems, USYMLQ, and therefore USYMLQR, diverges much as in the same way as CRAIG or SYMMLQ would diverge.

7. Extension: tridiagonalization in elliptic norms. In this section, we focus on the general saddle-point system (1) and assume that the $\mathbf{x}$ part of the solution is naturally measured in a norm defined by the symmetric and positive-definite matrix $\mathbf{N}$. Consider the scaled formulation of (1)

$$\begin{bmatrix} \mathbf{M}^{-\frac{1}{2}} & \\ & \mathbf{N}^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & \end{bmatrix} \begin{bmatrix} \mathbf{M}^{-\frac{1}{2}} & \\ & \mathbf{N}^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{M}^{\frac{1}{2}}\mathbf{r} \\ \mathbf{N}^{\frac{1}{2}}\mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{M}^{-\frac{1}{2}}\mathbf{b} \\ \mathbf{N}^{-\frac{1}{2}}\mathbf{c} \end{bmatrix}. \tag{44}$$

We apply Algorithm 1 to the scaled operator $\mathbf{M}^{-\frac{1}{2}}\mathbf{A}\mathbf{N}^{-\frac{1}{2}}$ with initial vectors $\mathbf{M}^{-\frac{1}{2}}\mathbf{b}$ and $\mathbf{N}^{-\frac{1}{2}}\mathbf{c}$ and perform the change of variable $\mathbf{u}_k \leftarrow \mathbf{M}^{-\frac{1}{2}}\mathbf{u}_k$ and $\mathbf{v}_k \leftarrow \mathbf{N}^{-\frac{1}{2}}\mathbf{v}_k$, and obtain Algorithm 3.

Algorithm 3 Saunders-Simon-Yip Tridiagonalization in Elliptic Norms

Require: $\mathbf{A}$, $\mathbf{b}$, $\mathbf{c}$, $\mathbf{M}^{-1}$, $\mathbf{N}^{-1}$

1: $\beta_1\mathbf{M}\mathbf{u}_1 = \mathbf{b}$, and $\gamma_1\mathbf{N}\mathbf{v}_1 = \mathbf{c}$, $(\beta_1, \gamma_1) > 0$ so that $\|\mathbf{u}_1\|_{\mathbf{M}} = \|\mathbf{v}_1\|_{\mathbf{N}} = 1$
2: **for** $k = 1, 2, \ldots$ **do**
3: $\quad \mathbf{q} = \mathbf{A}\mathbf{v}_k - \gamma_k\mathbf{M}\mathbf{u}_{k-1}$, $\alpha_k = \mathbf{u}_k^T\mathbf{q}$
4: $\quad \beta_{k+1}\mathbf{M}\mathbf{u}_{k+1} = \mathbf{q} - \alpha_k\mathbf{M}\mathbf{u}_k$, $\beta_{k+1} > 0$ so that $\|\mathbf{u}_{k+1}\|_{\mathbf{M}} = 1$
5: $\quad \gamma_{k+1}\mathbf{N}\mathbf{v}_{k+1} = \mathbf{A}^T\mathbf{u}_k - \beta_k\mathbf{N}\mathbf{v}_{k-1} - \alpha_k\mathbf{N}\mathbf{v}_k$, $\gamma_{k+1} > 0$ so that $\|\mathbf{v}_{k+1}\|_{\mathbf{N}} = 1$
6: **end for**

Line 1 of Algorithm 3 is compact notation for the sequence of operations

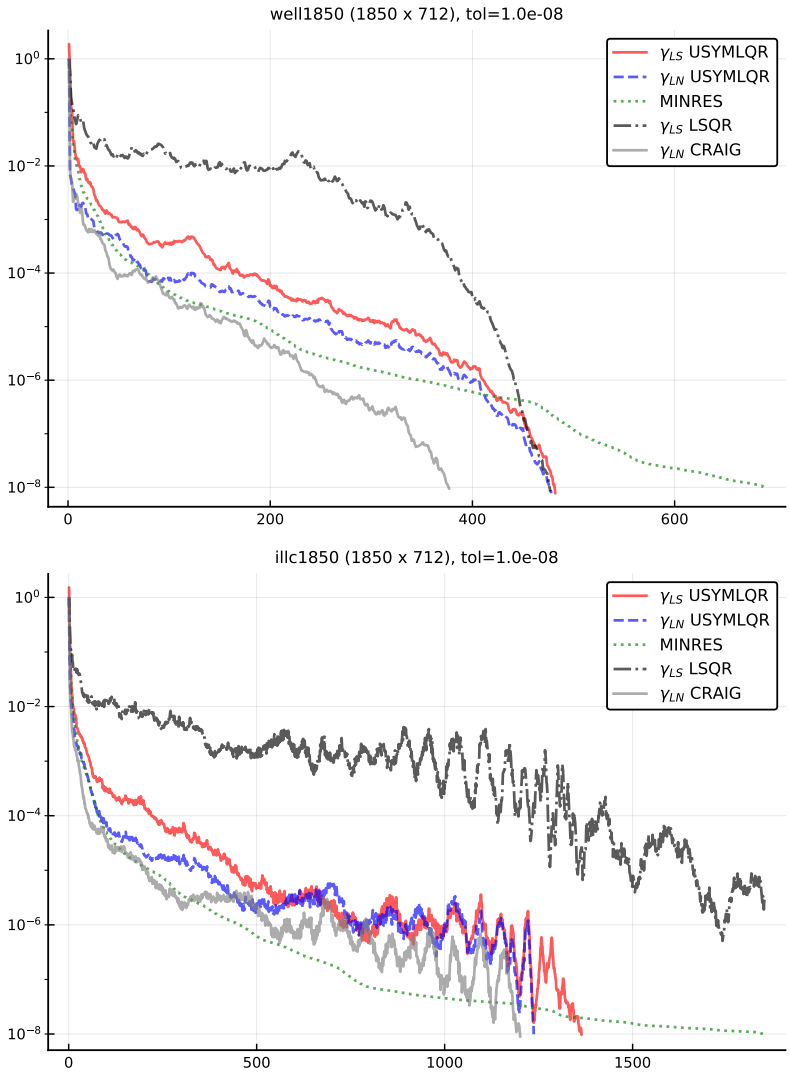

FIG. 1. *Convergence curves on under-determined problems from the SuiteSparse Matrix Collection. The horizontal axis represents iterations. In red and blue are the backward errors (40) and (42), respectively, along the* USYMLQR *iterations. The green dotted curve is the backward error (43) for (4) along the* MINRES *iterations. The black dash-dotted curve is the backward error (40) along* LSQR *iterations, while the gray curve is the backward error (42) along the* CRAIG *iterations.*

1. solve $\mathbf{M}\mathbf{u}_1 = \mathbf{b}$;
2. compute $\beta_1 = (\mathbf{u}_1^T\mathbf{b})^{\frac{1}{2}}$;
3. scale $\mathbf{u}_1 \leftarrow \mathbf{u}_1/\beta_1$ if $\beta_1 \neq 0$,

and similarly for $\mathbf{v}_1$. Lines 4–5 are similar.

Algorithm 3 generates matrices $\mathbf{U}_k$ and $\mathbf{V}_k$ that are $\mathbf{M}$- and $\mathbf{N}$-orthgonal, respectively. The process is characterized by the identities

$$\mathbf{A}\mathbf{V}_k = \mathbf{M}\mathbf{U}_k\mathbf{T}_k + \beta_{k+1}\mathbf{M}\mathbf{u}_{k+1}\mathbf{e}_k^T = \mathbf{M}\mathbf{U}_{k+1}\mathbf{T}_{k+1,k} \tag{45a}$$

$$\mathbf{A}^T\mathbf{U}_k = \mathbf{N}\mathbf{V}_k\mathbf{T}_k^T + \gamma_{k+1}\mathbf{N}\mathbf{v}_{k+1}\mathbf{e}_k^T = \mathbf{N}\mathbf{V}_{k+1}\mathbf{T}_{k,k+1}^T. \tag{45b}$$

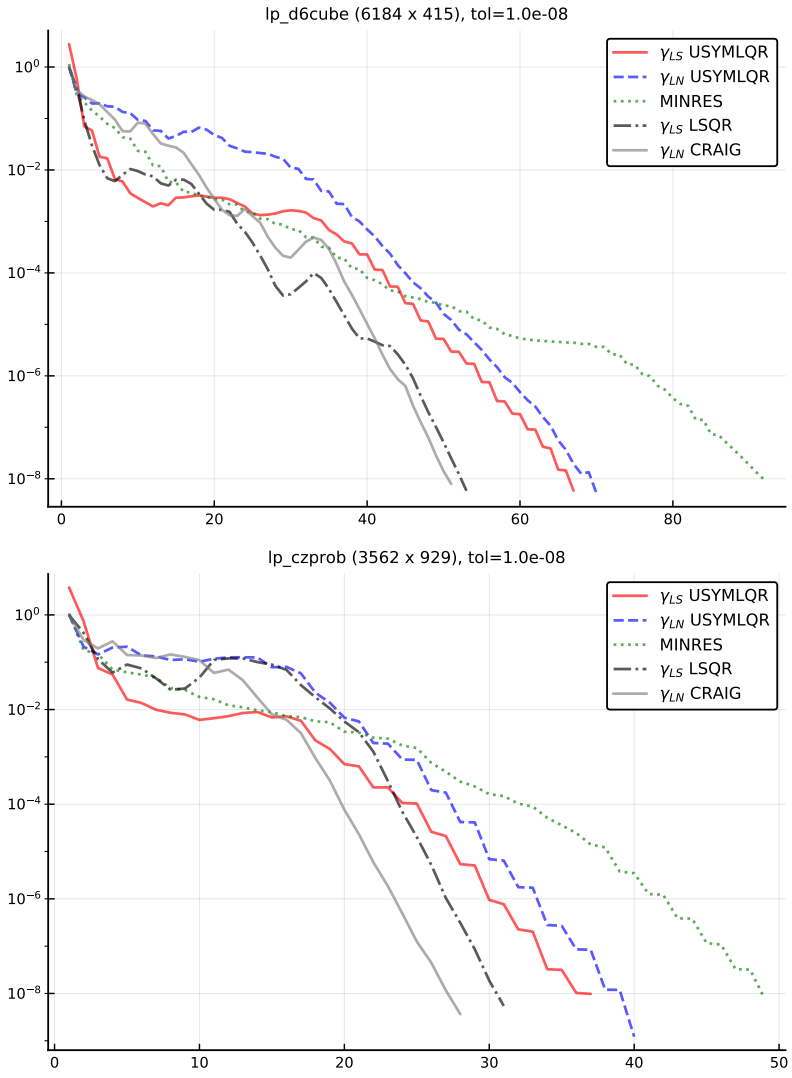

FIG. 2. *Convergence curves on under-determined problems from the SuiteSparse Matrix Collection. The horizontal axis represents iterations. In red and blue are the backward errors* (40) *and* (42)*, respectively, along the* USYMLQR *iterations. The green dotted curve is the backward error* (43) *for* (4) *along the* MINRES *iterations. The black dash-dotted curve is the backward error* (40) *along* LSQR *iterations, while the gray curve is the backward error* (42) *along the* CRAIG *iterations.*

Provided systems with matrices $\mathbf{M}$ and $\mathbf{N}$ can be solved efficiently at each iteration, it suffices to replace Algorithm 1 with Algorithm 3 to solve (1) with the USYMQR/USYMLQ combination. An example such situation occurs in certain regularization methods for constrained optimization, where $\mathbf{N}$ is typically a multiple of the identity and $\mathbf{M}$ is a limited-memory quasi-Newton approximation whose inverse can be applied efficiently—see, e.g., (Arreckx and Orban, 2018).

Above, $\mathbf{N}$ may be viewed as a preconditioner as it preserves the zero bottom block of (1), and can be chosen to cluster the singular values of $\mathbf{M}^{-\frac{1}{2}}\mathbf{A}\mathbf{N}^{-\frac{1}{2}}$. However, we do not favor this interpretation in terms of preconditioner as it does not persist

in the presence of regularization, where both $\mathbf{M}$ and $\mathbf{N}$ define the norms in which the least-squares and least-norm residuals should be measured, and those norms are typically defined by the user beforehand.

Golub et al. (2008) futher Saunders, Simon, and Yip's interpretation of Algorithm 1 as a block Lanczos method on an augmented system and their results elegantly carry over to the present framework. Pasting (45) together results in

$$\begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & \end{bmatrix} \begin{bmatrix} \mathbf{U}_k & \\ & \mathbf{V}_k \end{bmatrix} = \begin{bmatrix} \mathbf{M} & \\ & \mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{U}_k & \\ & \mathbf{V}_k \end{bmatrix} \begin{bmatrix} \mathbf{I}_k & \mathbf{T}_k \\ \mathbf{T}_k^T & \end{bmatrix} + \begin{bmatrix} \beta_{k+1} \mathbf{M} \mathbf{u}_{k+1} \mathbf{e}_k^T \\ \gamma_{k+1} \mathbf{N} \mathbf{v}_{k+1} \mathbf{e}_{2k}^T \end{bmatrix}, \tag{46}$$

i.e., a block-lanczos process applied to the operator of (4) in the norm defined by $\text{blkdiag}(\mathbf{M}, \mathbf{N})$. The Lanczos vectors have the form $(\mathbf{u}_k, \mathbf{0})$ or $(\mathbf{0}, \mathbf{v}_k)$. The permutation

$$\boldsymbol{\Pi} := \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_{k+1} & \mathbf{e}_2 & \mathbf{e}_{k+2} & \ldots & \mathbf{e}_k & \mathbf{e}_{2k} \end{bmatrix},$$

introduced by Paige (1974) restores the order in which Algorithm 3 generates them:

$$\begin{bmatrix} \mathbf{U}_k & \\ & \mathbf{V}_k \end{bmatrix} \boldsymbol{\Pi}^T = \begin{bmatrix} \mathbf{u}_1 & \mathbf{0} & \mathbf{u}_2 & \ldots & \mathbf{u}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{v}_1 & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{v}_k \end{bmatrix}.$$

The permutation $\boldsymbol{\Pi}$ also shuffles the small symmetric saddle-point operator in the right-hand side of (46) to block tridiagonal form with blocks of size 2:

$$\boldsymbol{\Pi} \begin{bmatrix} \mathbf{I}_k & \mathbf{T}_k \\ \mathbf{T}_k^T & \end{bmatrix} \boldsymbol{\Pi}^T = \begin{bmatrix} \boldsymbol{\alpha}_1 & \boldsymbol{\beta}_2^T & & \\ \boldsymbol{\beta}_2 & \boldsymbol{\alpha}_2 & \ddots & \\ & \ddots & \ddots & \boldsymbol{\beta}_{k-1}^T \\ & & \boldsymbol{\beta}_{k-1} & \boldsymbol{\alpha}_k \end{bmatrix}, \qquad \boldsymbol{\alpha}_j := \begin{bmatrix} 1 & \alpha_j \\ \alpha_j & \end{bmatrix}, \quad \boldsymbol{\beta}_{j+1} := \begin{bmatrix} 0 & \beta_{j+1} \\ \gamma_j & 0 \end{bmatrix}.$$

As a result of this block-Lanczos interpretation USYMLQR sometimes terminates in about half as many iterations as MINRES. Figure 3 illustrates convergence curves on such a problem.

We implemented Algorithm 3 as a generalization of Algorithm 1. Only the basis-generation process is affected by the change and the updated implementation requires extra storage for vectors $\mathbf{Mu}$ and $\mathbf{Nv}$ at iterations $k-1$, k, and $k+1$. We illustrate the behavior of the backward errors (39) and (41) and compare them to (43) generated by MINRES with preconditioner $\text{blkdiag}(\mathbf{M}, \mathbf{N})$. Our test systems were generated during the iterations of an interior-point method for convex quadratic optimization and are described by Orban (2015a). The quadratic problems originate from the CUTEst collection (Gould, Orban, and Toint, 2015). All systems are available in MatrixMarket format (Orban, 2015b) and have the form (4) with $\mathbf{M} = \mathbf{H} + \mathbf{D} + \rho\mathbf{I}$, where $\mathbf{H}$ is the Hessian of the objective, $\mathbf{D}$ is diagonal and positive semi-definite, and $\rho > 0$ is a regularization parameter. The leading block $\mathbf{M}$ becomes increasingly ill conditioned as the interior-point iteration counter grows. In each experiment, we select $\mathbf{N} = \mathbf{I}$. For our current purpose of illustrating Algorithm 3, we precompute the Cholesky factorization of $\mathbf{M}$ prior to calling USYMLQR, and perform a forward and a backsolve each time applying $\mathbf{M}^{-1}$ is requested.

Figure 4 illustrates the behavior of USYMLQR, LSQR, CRAIG and MINRES on problems `primalc1` and `dualc1` at interior-point iterations 0, 5 and 10. The elliptic-norm variants of LSQR and CRAIG are as described by Orban and Arioli (2017). Note that as the interior-point iteration counter increases, the convergence curves

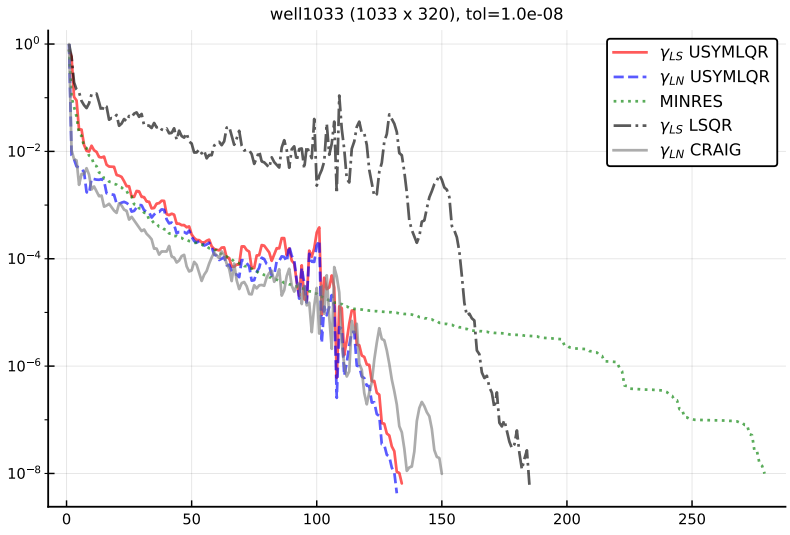

FIG. 3. *Convergence curves on an over-determined problems from the SuiteSparse Matrix Collection. The horizontal axis represents iterations. In red and blue are the backward errors* (40) *and* (42)*, respectively, along the* USYMLQR *iterations. The green dotted curve is the backward error* (43) *for* (4) *along the* MINRES *iterations. The black dash-dotted curve is the backward error* (40) *along* LSQR *iterations, while the gray curve is the backward error* (42) *along the* CRAIG *iterations.*

of USYMLQR, LSQR and CRAIG become progressively more oscillatory. On `primalc1`, USYMLQR and MINRES perform comparably at interior-point iterations 0 and 5, but the convergence of USYMLQR deteriorates at iteration 10, probably due to the increasing ill conditioning of $\mathbf{M}$. On `dualc1`, we set the tolerance to 1.0e−7 at iteration 0. USYMLQR manages to decrease both (39) and (41) below 1.0e−7, The measure (39) drops below 1.0e−8 after an additional iteration, but (39) diverges soon after. By contrast, the MINRES residual drops below 1.0e−8 after 15 iterations. USYMLQR terminates earlier than MINRES at interior-point iterations 5 and 10. In all cases however, USYMLQR performs comparably to LSQR and CRAIG but has the advantage of solving both problems in (3) simultaneously.

8. Conclusion. Contrary to the Golub and Kahan bidiagonalization and Lanczos processes, the orthogonal tridiagonalization of Saunders et al. requires two initial vectors. This distinguishing feature makes it particularly suited to the solution of symmetric saddle-point systems with a positive definite leading block. Thanks to an appropriate decomposition of the saddle-point system into a least-squares and a least-norm problem, it is possible to solve the system in one pass by combining the solutions of the two problems, which can be solved concurrently. An appropriate structured backward-error analysis provides stopping criteria for the least-squares and least-norm problems guaranteeing that the combined solution is backward stable for (4). The overall storage and computational effort is comparable to that of MINRES.

A side benefit of the present research is to provide a numerical evaluation of USYMQR and USYMLQ on rectangular problems, as they had so far only been run on square problems in the literature.

USYMLQR is closer to SYMMLQ than to MINRES in that it is only well defined for consistent systems and will stagnate or diverge on inconsistent systems. Despite the fact that only the USYMQR least-squares residual and USYMLQ error norm are

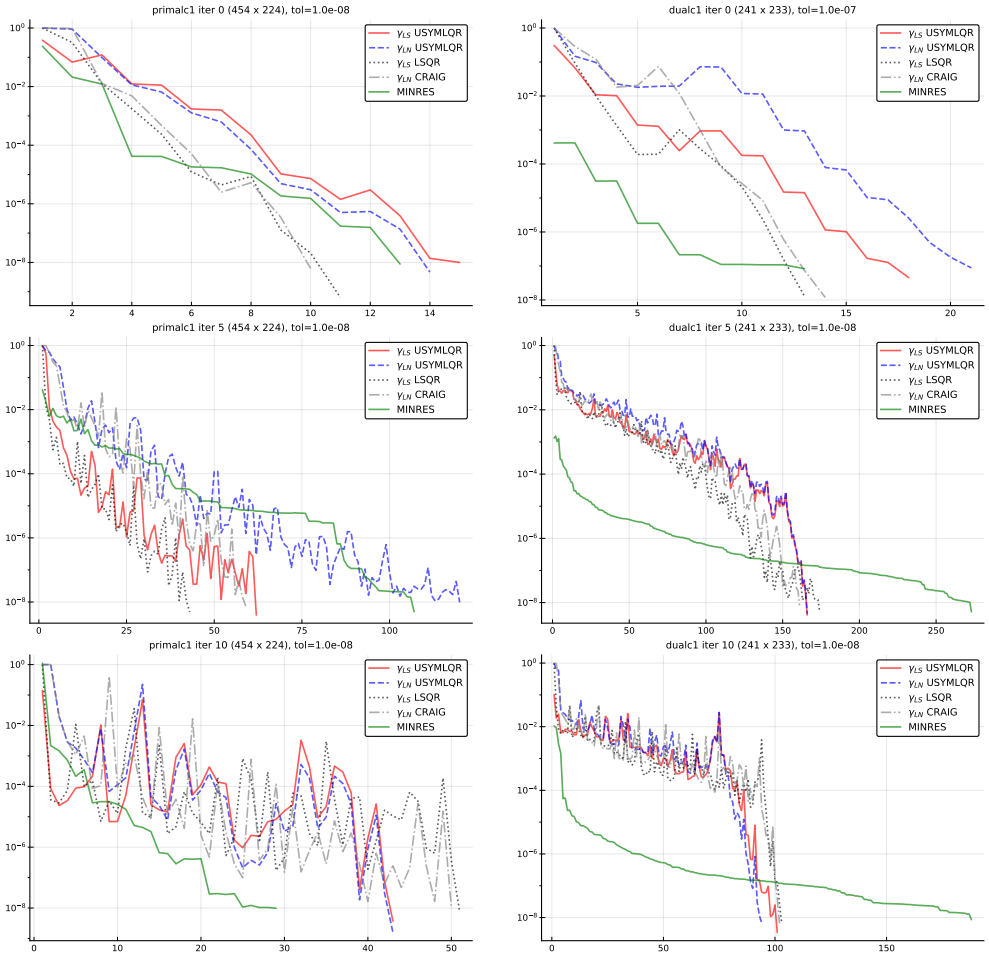

FIG. 4. *Convergence curves on problems* ***primalc1*** *(left) and* ***dualc1*** *(right) from the* ***CUTEst*** *collection. The horizontal axis represents iterations. In red and blue are the backward errors* (40) *and* (42)*, respectively, along the* USYMLQR *iterations. The green dotted curve is the backward error* (43) *for* (4) *with block-diagonal preconditioner along the* MINRES *iterations.*

monotonic, USYMLQR is attractive as it may converge in fewer iterations than MINRES.

References.

W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quart. Appl. Math.*, 9:17–29, 1951.

S. Arreckx and D. Orban. A regularized factorization-free method for equality-constrained optimization. *SIAM J. Optim.*, 28(2):1613–1639, 2018. DOI: 10.13140/RG.2.2.20368.00007.

M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005. DOI: 10.1017/S0962492904000212.

J. E. Craig. The N-step iteration procedures. *J. Math. and Physics*, 34(1):64–73, 1955. DOI: 10.1002/sapm195534164.

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

D. C.-L. Fong and M. A. Saunders. LSMR: An iterative algorithm for sparse least-squares problems. *SIAM J. Sci. Comput.*, 33(5):2950–2971, 2011. DOI: 10.1137/10079687X.

G. H. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix.

SIAM J. Numer. Anal., 2(2):205–224, 1965. DOI: 10.1137/0702016.

G. H. Golub, M. Stoll, and A. Wathen. Approximation of the scattering amplitude and linear systems. *Electron. Trans. Numer. Anal.*, 31:178–203, 2008.

N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEst: a Constrained and Unconstrained Testing Environment with safe threads for Mathematical Optimization. *Computational Optimization and Applications*, 60:545–557, 2015. DOI: 10.1007/s10589-014-9687-3.

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

C. Lanczos. Solution of systems of linear equations by minimized iterations. *J. Res. Nat. Bur. Standards*, 49(1):33–53, 1952.

D. Orban. A collection of linear systems arising from interior-point methods for quadratic optimization. Cahier du GERAD G-2015-117, GERAD, Montréal, Canada, 2015a.

D. Orban. A collection of linear systems arising from interior-point methods for quadratic optimization. Online data set. DOI: 10.5281/zenodo.34130, 2015b.

D. Orban and M. Arioli. *Iterative Solution of Symmetric Quasi-Definite Linear Systems*, volume 3 of *Spotlights*. SIAM, 2017. DOI: 10.1137/1.9781611974737.

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

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

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

L. Reichel and Q. Ye. A generalized LSQR algorithm. *Numer. Linear Algebra Appl.*, 15(7): 643–660, 2008. DOI: 10.1002/nla.611.

J. L. Rigal and J. Gaches. On the compatibility of a given solution with the data of a linear system. *J. ACM*, 14(3):543–548, July 1967. ISSN 0004-5411. DOI: 10.1145/321406.321416. URL http://doi.acm.org/10.1145/321406.321416.

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

J.-G. Sun. Structured backward errors for KKT systems. *Linear Algebra Appl.*, 288:75–88, 1999. DOI: 10.1016/S0024-3795(98)10184-2.

H. Xiang and Y. Wei. On normwise structured backward errors for saddle point systems. *SIAM J. Matrix Anal. Appl.*, 29(3):838–849, 2007. DOI: 10.1137/060663684.

List of changes. *List of changes is available after the next LaTeX run.*

Commit 023009c by Dominique Orban on 2015-10-19 16:46:34 -0400