SIAM J. Matrix Anal. Appl.
Vol. 41, No. 3, pp. 1145–1166

BILQ: AN ITERATIVE METHOD FOR NONSYMMETRIC LINEAR SYSTEMS WITH A QUASI-MINIMUM ERROR PROPERTY*

ALEXIS MONTOISON† AND DOMINIQUE ORBAN†

Abstract. We introduce an iterative method named BiLQ for solving general square linear systems $Ax = b$ that is based on the Lanczos biorthogonalization process defined by least-norm subproblems and that is a natural companion to BiCG and Qmr. Whereas the iterates of BiCG, Cgs, and BiCGStab may not exist when the tridiagonal projection of A is singular, BiLQ is reliable on compatible systems even if A is ill-conditioned or singular. As in the symmetric case, the BiCG residual is often smaller than the BiLQ residual, and when the BiCG iterate exists, an inexpensive transfer from the BiLQ iterate is possible. Although the Euclidean norm of the BiLQ error is usually not monotonic, it is monotonic in a different norm that depends on the Lanczos vectors. We establish a similar property for the Qmr residual. BiLQ combines with Qmr to take advantage of two initial vectors and solve a system and an adjoint system simultaneously at a cost similar to that of applying either method. We derive an analogous combination of Usymlq and Usymqr based on the orthogonal tridiagonalization process. The resulting combinations, named BiLQR and TriLQR, may be used to estimate integral functionals involving the solution of a primal and an adjoint system. We compare BiLQR and TriLQR with Minres-qlp on a related augmented system, which performs a comparable amount of work and requires comparable storage. In our experiments, BiLQR terminates earlier than TriLQR and Minres-qlp in terms of residual and error of the primal and adjoint systems.

Key words. least-norm subproblems, Lanczos biorthogonalization process, adjoint systems, integral functional, orthogonal, tridiagonalization process, quasi-minimum error method, iterative methods, multiprecision

AMS subject classifications. 15A06, 65F10, 65F25, 65F50, 93E24, 90C06

DOI. 10.1137/19M1290991

1. Introduction.

We consider the square consistent linear system

$$Ax = b, \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$ can be nonsymmetric, is either large and sparse, or is only available as a linear operator, i.e., via operator-vector products. Systems such as (1.1) arise in the discretization of PDEs in numerous applications, including compressible turbulent fluid flow (Chisholm and Zingg, 2009) and circuit simulation (Davis and Natarajan, 2012). We consider Krylov subspace methods and are interested in generating iterates with guaranteed decrease of the error $x_k - x_\star$ in a certain norm, where $x_\star$ is the solution of (1.1).

The foundation of Krylov methods is a basis-generation process upon which three methods may be developed: one computing the minimum-norm solution of an underdetermined system, one solving a square system and imposing a Galerkin condition, and one solving an overdetermined system in the least-squares sense. These methods may be implemented with the help of LQ, LU, or QR factorizations of a related operator, respectively.

[*] Received by the editors October 3, 2019; accepted for publication (in revised form) April 8, 2020; published electronically August 4, 2020.
https://doi.org/10.1137/19M1290991

Funding: The work of the first author was partially supported by merit and excellence scholarships from the Arbor Foundation and the IVADO Institute. The work of the second author was partially supported by an NSERC Discovery grant.

[†] GERAD and Department of Mathematics and Industrial Engineering, Polytechnique Montréal, H3T 1J4 QC Canada (alexis.montoison@polymtl.ca, dominique.orban@gerad.ca).

In this paper, we develop an iterative method named BiLQ of the first type based on the Lanczos (1950) biorthogonalization process. Together with BiCG (Fletcher, 1976) and Qmr (Freund and Nachtigal, 1991), BiLQ completes the family of methods based on the biorthogonalization process. Our motivation for developing BiLQ is that the BiCG iterate is not necessarily well defined at each iteration. By contrast, the BiLQ iterate is always determined from a well-posed subproblem. In addition, whenever the BiCG iterate is well defined, it is possible to transition from the BiLQ iterate to the BiCG iterate if desired. In this sense, it is always safe to use BiLQ in place of BiCG. We begin by stating the defining properties of BiLQ, describing its implementation in detail, and illustrating its behavior on numerical examples side by side with BiCG and Qmr.

In a second stage, we exploit the fact that the biorthogonalization process requires two initial vectors to develop a combination of BiLQ and Qmr that solves (1.1) together with a dual system

$$A^T t = c \tag{1.2}$$

simultaneously at a cost comparable to that of applying BiLQ or Qmr to solve only one of those systems. The resulting combination is named BiLQR and is employed to illustrate the computation of superconvergent estimates of integral functionals arising in certain PDE problems.

We note that a similar approach may be developed for the Saunders, Simon, and Yip (1988) orthogonal tridiagonalization process, which also requires two initial vectors, by combining Usymlq and Usymqr. The resulting combination is named TriLQR.

Finally, we compare BiLQR and TriLQR with Minres-qlp on a related augmented system to solve both (1.1) and (1.2) simultaneously. In our experiments, BiLQR terminates earlier than TriLQR and Minres-qlp in terms of residual and error of the primal and adjoint systems. BiLQR and TriLQR are also cheaper to apply than the methods of Lu and Darmofal (2003) and Golub, Stoll, and Wathen (2008), as discussed in the next section.

Our Julia (Bezanson et al., 2017) implementations of BiLQ, Qmr, Usymlq, Usymqr, BiLQR, TriLQR, and Minres-qlp are available from https://github.com/JuliaSmoothOptimizers/Krylov.jl. Thanks to multiple dispatch, a language feature allowing automatic compilation of variants of each method corresponding to inputs expressed in various floating-point systems, our implementations run in any floating-point precision supported.

Related research. Paige and Saunders (1975) developed one of the best-known minimum error methods, Symmlq, based on the symmetric Lanczos process. Symmlq inspired Estrin, Orban, and Saunders (2019a,b) to develop Lslq and Lnlq for rectangular problems based on the Golub and Kahan (1965) process. Lslq and Lnlq are equivalent to Symmlq applied to the normal equations and normal equations of the second kind, respectively.

Saunders, Simon, and Yip (1988) define Usymlq for square consistent systems based on the orthogonal tridiagonalization process. Usymlq is based on a subproblem similar to that of Symmlq and coincides with Symmlq in the symmetric case. Its companion method, Usymqr, is similar in spirit to Minres. Buttari et al. (2019) combine both into a method named Usymlqr designed to solve symmetric saddle-point systems with general right-hand side and inspire the development of BiLQR and TriLQR in the present paper.

Weiss (1994) describes two types of error-minimizing Krylov methods for square A: one based on a process applied to A^TA and one to A^T. Our approach is to apply the biorthogonalization process directly to A. We defer a numerical stability analysis to future work but note that Paige, Panayotov, and Zemke (2014) study the augmented stability of the biorthogonalization process. In this sense, we make the implicit assumption that computations are carried out in exact arithmetic. This assumption prompted us to develop our implementations so that they can be applied in any supported floating-point arithmetic.

The simultaneous solution of a system and an adjoint system has attracted attention in the past. Notably, Lu and Darmofal (2003) devise a variant of Qmr to solve both systems at once at a cost approximately equal to that of Qmr applied to one of the systems but with an increase in storage requirements. Golub, Stoll, and Wathen (2008) follow a similar approach and use a variant of Usymqr to solve both (1.1) and (1.2). An advantage of Usymqr is to produce monotonic residuals in the Euclidean norm for both systems. We illustrate in Table 2 that our methods are cheaper and have smaller storage requirements than those of Lu and Darmofal (2003) and Golub, Stoll, and Wathen (2008), though residuals are not monotonic in the Euclidean norm.

Notation. Matrices and vectors are denoted by uppercase and lowercase Latin letters, respectively, and scalars by Greek letters. An exception is made for cosines and sines (c, s) that compose Givens reflections. For a vector v, $\|v\|$ denotes the Euclidean norm of v, and for symmetric and positive-definite N, the N-norm of v is $\|v\|_N^2 = v^TNv$. For a matrix M, $\|M\|_F$ denotes the Frobenius norm of M. The vector e_i is the ith column of an identity matrix of size dictated by the context. Vectors and scalars decorated by a bar will be updated at the next iteration. For $j = 2, \dots, k$, we use the compact representation

$$Q_{j-1,j} = \begin{array}{c} \begin{array}{cc} {\scriptstyle j-1} & {\scriptstyle j} \end{array} \\ \begin{bmatrix} c_j & s_j \\ s_j & -c_j \end{bmatrix} \end{array} := \begin{bmatrix} I_{j-2} & & & \\ & c_j & s_j & \\ & s_j & -c_j & \\ & & & I_{k-j} \end{bmatrix}$$

for orthogonal reflections, where $s_j^2 + c_j^2 = 1$, where border indices indicate row and column numbers, and where I_k represents the $k \times k$ identity operator. We abuse the notation $\bar{z}_k = (z_{k-1},\ \bar{\zeta}_k)$ to represent the column vector $\begin{bmatrix} z_{k-1}^T & \bar{\zeta}_k \end{bmatrix}^T$.

2. Derivation of BiLQ.

2.1. The Lanczos biorthogonalization process.

The Lanczos biorthogonalization process generates sequences of vectors $\{v_k\}$ and $\{u_k\}$ such that $v_i^Tu_j = \delta_{ij}$ in exact arithmetic for as long as the process does not break down. The process is summarized as Algorithm 2.1.

We denote $V_k = \begin{bmatrix} v_1 & \dots & v_k \end{bmatrix}$ and $U_k = \begin{bmatrix} u_1 & \dots & u_k \end{bmatrix}$. We choose the scaling factors β_k and γ_k so that $v_k^Tu_k = 1$ for all $k \geq 1$, i.e., $V_k^TU_k = I_k$. After k iterations, the situation may be summarized as

$$AV_k = V_kT_k \ + \beta_{k+1}v_{k+1}e_k^T = V_{k+1}T_{k+1,k}, \tag{2.1a}$$

$$A^TU_k = U_kT_k^T + \gamma_{k+1}u_{k+1}e_k^T = U_{k+1}T_{k,k+1}^T, \tag{2.1b}$$

Algorithm 2.1 Lanczos Biorthogonalization Process.

```
Require: $A$, $b$, $c$
$v_0 = 0$, $u_0 = 0$
$\beta_1 v_1 = b$, $\gamma_1 u_1 = c$                    $(\beta_1, \gamma_1)$ so that $v_1^T u_1 = 1$
for $k = 1, 2, \ldots$ do
    $q = Av_k - \gamma_k v_{k-1}$, $\alpha_k = u_k^T q$
    $p = A^T u_k - \beta_k u_{k-1}$
    $\beta_{k+1} v_{k+1} = q - \alpha_k v_k$                $(\beta_{k+1}, \gamma_{k+1})$ so that $v_{k+1}^T u_{k+1} = 1$
    $\gamma_{k+1} u_{k+1} = p - \alpha_k u_k$
end for
```

where

$$T_k = \begin{bmatrix} \alpha_1 & \gamma_2 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \gamma_k \\ & & \beta_k & \alpha_k \end{bmatrix}, \; T_{k,k+1} = \begin{bmatrix} T_k & \gamma_{k+1} e_k \end{bmatrix}, \; T_{k+1,k} = \begin{bmatrix} T_k \\ \beta_{k+1} e_k^T \end{bmatrix}.$$

The columns of V_k and U_k form a basis for $\mathcal{K}_k := \text{Span}\{b, Ab, \ldots, A^{k-1}b\}$ and $\mathcal{L}_k := \text{Span}\{c, A^T c, \ldots, (A^T)^{k-1} c\}$, respectively. Though V_k cannot be expected to be orthogonal to U_k in inexact arithmetic, and therefore $U_k^T A V_k = T_k$ cannot be expected to hold, (2.1) holds to within machine precision.

2.2. Definition of BiLQ.

BiCG (Fletcher, 1976) generates an approximation $x_k^{\text{C}} = V_k y_k^{\text{C}}$ to a solution of (1.1), where $y_k^{\text{C}} \in \mathbb{R}^k$ solves

$$T_k y = \beta_1 e_1. \tag{2.2}$$

Unfortunately, T_k could be singular, and (2.2) could be inconsistent or have multiple solutions. In such situations, x_k^{C} is undefined.

To remedy such situations, we define BiLQ as the method that generates an approximation $x_k^{\text{L}} = V_k y_k^{\text{L}}$, where $y_k^{\text{L}} \in \mathbb{R}^k$ solves

$$\underset{y}{\text{minimize}} \;\; \|y\| \quad \text{subject to} \;\; T_{k-1,k} y = \beta_1 e_1. \tag{2.3}$$

Note that the constraints of (2.3) are consistent as long as Algorithm 2.1 does not break down so that x_k^{L} is always well defined.

Qmr (Freund and Nachtigal, 1991) generates $x_k^{\text{Q}} = V_k y_k^{\text{Q}}$, where $y_k^{\text{Q}} \in \mathbb{R}^k$ solves

$$\underset{y}{\text{minimize}} \;\; \|T_{k+1,k} y - \beta_1 e_1\|, \tag{2.4}$$

so that x_k^{Q} is also always well defined.

When A is symmetric and $b = c$, Algorithm 2.1 coincides with the symmetric Lanczos process, and the three above methods are equivalent to Cg (Hestenes and Stiefel, 1952), Symmlq (Paige and Saunders, 1975), and Minres (Paige and Saunders, 1975), respectively.

2.3. An LQ factorization. We determine y_k^{L}, the solution of (2.3), via the LQ factorization of $T_{k-1,k}$, which we obtain from the LQ factorization

$$T_k = \bar{L}_k Q_k, \text{ where} \tag{2.5a}$$

$$\overline{L}_k = \begin{bmatrix} \delta_1 & & & & & \\ \lambda_1 & \delta_2 & & & & \\ \varepsilon_1 & \lambda_2 & \delta_3 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \varepsilon_{k-3} & \lambda_{k-2} & \delta_{k-1} & \\ & & & \varepsilon_{k-2} & \lambda_{k-1} & \bar{\delta}_k \end{bmatrix} = \begin{bmatrix} L_{k-1} & 0 \\ \varepsilon_{k-2} e_{k-2}^T + \lambda_{k-1} e_{k-1}^T & \bar{\delta}_k \end{bmatrix} \tag{2.5b}$$

and $Q_k^T = Q_{1,2} Q_{2,3} \cdots Q_{k-1,k}$ is orthogonal and defined as a product of Givens reflections. Indeed, the above yields the LQ factorization

$$T_{k-1,k} = \begin{bmatrix} L_{k-1} & 0 \end{bmatrix} Q_k. \tag{2.6}$$

If we initialize $\bar{\delta}_1 := \alpha_1$, $\bar{\lambda}_1 := \beta_2$, $c_1 = -1$, and $s_1 = 0$, individual factorization steps may be represented as an application of $Q_{k-2,k-1}$ to $T_k Q_{k-2}^T$,

$$\begin{array}{c} \\ {\scriptstyle k-2} \\ {\scriptstyle k-1} \\ {\scriptstyle k} \end{array} \begin{array}{c} \begin{array}{ccc} {\scriptstyle k-2} & {\scriptstyle k-1} & {\scriptstyle k} \end{array} \\ \begin{bmatrix} \bar{\delta}_{k-2} & \gamma_{k-1} & \\ \bar{\lambda}_{k-2} & \alpha_{k-1} & \gamma_k \\ & \beta_k & \alpha_k \end{bmatrix} \end{array} \begin{array}{c} \begin{array}{ccc} {\scriptstyle k-2} & {\scriptstyle k-1} & {\scriptstyle k} \end{array} \\ \begin{bmatrix} c_{k-1} & s_{k-1} & \\ s_{k-1} & -c_{k-1} & \\ & & 1 \end{bmatrix} \end{array} = \begin{array}{c} \begin{array}{ccc} {\scriptstyle k-2} & {\scriptstyle k-1} & {\scriptstyle k} \end{array} \\ \begin{bmatrix} \delta_{k-2} & 0 & \\ \lambda_{k-2} & \bar{\delta}_{k-1} & \gamma_k \\ \varepsilon_{k-2} & \bar{\lambda}_{k-1} & \alpha_k \end{bmatrix} \end{array},$$

followed by an application of $Q_{k-1,k}$ to the result:

$$\begin{array}{c} \\ {\scriptstyle k-2} \\ {\scriptstyle k-1} \\ {\scriptstyle k} \end{array} \begin{array}{c} \begin{array}{ccc} {\scriptstyle k-2} & {\scriptstyle k-1} & {\scriptstyle k} \end{array} \\ \begin{bmatrix} \delta_{k-2} & & \\ \lambda_{k-2} & \bar{\delta}_{k-1} & \gamma_k \\ \varepsilon_{k-2} & \bar{\lambda}_{k-1} & \alpha_k \end{bmatrix} \end{array} \begin{array}{c} \begin{array}{ccc} {\scriptstyle k-2} & {\scriptstyle k-1} & {\scriptstyle k} \end{array} \\ \begin{bmatrix} 1 & & \\ & c_k & s_k \\ & s_k & -c_k \end{bmatrix} \end{array} = \begin{array}{c} \begin{array}{ccc} {\scriptstyle k-2} & {\scriptstyle k-1} & {\scriptstyle k} \end{array} \\ \begin{bmatrix} \delta_{k-2} & & \\ \lambda_{k-2} & \delta_{k-1} & \\ \varepsilon_{k-2} & \lambda_{k-1} & \bar{\delta}_k \end{bmatrix} \end{array}.$$

The reflection $Q_{k-1,k}$ is designed to zero out γ_k on the superdiagonal of T_k and affects three rows and two columns, It is defined by

$$\delta_{k-1} = \sqrt{\bar{\delta}_{k-1}^2 + \gamma_k^2}, \quad c_k = \bar{\delta}_{k-1}/\delta_{k-1}, \quad s_k = \gamma_k/\delta_{k-1} \tag{2.7}$$

and yields the recursion

$$\varepsilon_{k-2} = s_{k-1}\beta_k, \qquad k \geq 3, \tag{2.8a}$$

$$\bar{\lambda}_{k-1} = -c_{k-1}\beta_k, \qquad k \geq 3, \tag{2.8b}$$

$$\lambda_{k-1} = c_k\bar{\lambda}_{k-1} + s_k\alpha_k, \quad k \geq 2, \tag{2.8c}$$

$$\bar{\delta}_k = s_k\bar{\lambda}_{k-1} - c_k\alpha_k, \quad k \geq 2. \tag{2.8d}$$

2.4. Definition and update of the BiLQ and BiCG iterates. In order to compute the solution y_k^{L} of (2.3) using (2.6), we solve $\begin{bmatrix} L_{k-1} & 0 \end{bmatrix} Q_k y_k^{\text{L}} = \beta_1 e_1$. If $z_{k-1} := (\zeta_1, \ldots, \zeta_{k-1})$ is defined from $L_{k-1} z_{k-1} = \beta_1 e_1$, the minimum-norm solution of (2.3) is $y_k^{\text{L}} = Q_k^T \left[\begin{smallmatrix} z_{k-1} \\ 0 \end{smallmatrix}\right]$, and $\|y_k^{\text{L}}\| = \|z_{k-1}\|$.

We may compute y_k^{C} in (2.2) simultaneously as a cheap update of y_k^{L}. Indeed, (2.2) and (2.5) yield $\overline{L}_k Q_k y_k^{\text{C}} = \beta_1 e_1$. Let $\bar{z}_k := (z_{k-1}, \bar{\zeta}_k)$ be defined so that $\overline{L}_k \bar{z}_k = \beta_1 e_1$.

Then $y_k^{\mathrm{C}} = Q_k^T \bar{z}_k$. If $\bar{\delta}_k = 0$, y_k^{C} and the BiCG iterate x_k^{C} are undefined. The components of $\bar{z}_k$ are computed from

$$\eta_k = \begin{cases} \beta_1, & k = 1, \\ -\lambda_1 \zeta_1, & k = 2, \\ -\varepsilon_{k-2}\zeta_{k-2} - \lambda_{k-1}\zeta_{k-1}, & k \geq 3, \end{cases} \tag{2.9a}$$

$$\zeta_{k-1} = \eta_{k-1}/\delta_{k-1}, \quad k \geq 2, \tag{2.9b}$$

$$\bar{\zeta}_k = \eta_k/\bar{\delta}_k, \qquad \text{if } \bar{\delta}_k \neq 0. \tag{2.9c}$$

By definition, $x_k^{\mathrm{L}} = V_k y_k^{\mathrm{L}}$ and $x_k^{\mathrm{C}} = V_k y_k^{\mathrm{C}}$. To avoid storing V_k, we let

$$\overline{D}_k := V_k Q_k^T = \begin{bmatrix} d_1, & d_2, & \ldots, & d_{k-1}, & \bar{d}_k \end{bmatrix}, \quad \bar{d}_1 = v_1, \tag{2.10}$$

defined by the recursion

$$\begin{aligned} d_{k-1} &= c_k \bar{d}_{k-1} + s_k v_k \\ \bar{d}_k &= s_k \bar{d}_{k-1} - c_k v_k. \end{aligned} \tag{2.11}$$

Finally,

$$x_k^{\mathrm{L}} = V_k y_k^{\mathrm{L}} = \overline{D}_k \begin{bmatrix} z_{k-1} \\ 0 \end{bmatrix} = D_{k-1} z_{k-1} = x_{k-1}^{\mathrm{L}} + \zeta_{k-1} d_{k-1}, \tag{2.12a}$$

$$x_k^{\mathrm{C}} = V_k y_k^{\mathrm{C}} = \overline{D}_k \bar{z}_k = D_{k-1} z_{k-1} + \bar{\zeta}_k \bar{d}_k = x_k^{\mathrm{L}} + \bar{\zeta}_k \bar{d}_k. \tag{2.12b}$$

We see from (2.12b) that it is possible to transfer from x_k^{L} to x_k^{C} cheaply provided $\bar{\zeta}_k \neq 0$. Such transfer was described by Paige and Saunders (1975) as an inexpensive update from the Symmlq to the Cg point in the symmetric case.

2.5. Residual estimates.

The identity (2.1a) allows us to write the residual associated with $x_k = V_k y_k$ as

$$r_k = b - Ax_k = \beta_1 v_1 - AV_k y_k = \beta_1 v_1 - V_{k+1} T_{k+1,k} y_k.$$

Thus, (2.3) yields the residual at the BiLQ iterate:

$$\begin{aligned} r_k^{\mathrm{L}} &= V_{k-1}(\beta_1 e_1 - T_{k-1,k} y_k^{\mathrm{L}}) - (\beta_k e_{k-1} + \alpha_k e_k)^T y_k^{\mathrm{L}} v_k - \beta_{k+1} e_k^T y_k^{\mathrm{L}} v_{k+1} \\ &= -(\beta_k e_{k-1} + \alpha_k e_k)^T y_k^{\mathrm{L}} v_k - \beta_{k+1} e_k^T y_k^{\mathrm{L}} v_{k+1} \end{aligned} \tag{2.13}$$

and (2.2) yields the residual at the BiCG iterate:

$$r_k^{\mathrm{C}} = V_k(\beta_1 e_1 - T_k y_k^{\mathrm{C}}) - \beta_{k+1} v_{k+1} e_k^T y_k^{\mathrm{C}} = -\beta_{k+1} e_k^T y_k^{\mathrm{C}} v_{k+1}. \tag{2.14}$$

Because $Q_k^T = Q_{1,2} Q_{2,3} \cdots Q_{k-1,k}$, we have

$$\begin{aligned} e_{k-1}^T Q_k^T &= e_{k-1}^T Q_{k-2,k-1} Q_{k-1,k} = s_{k-1} e_{k-2}^T - c_{k-1} c_k e_{k-1}^T - c_{k-1} s_k e_k^T, \\ e_k^T Q_k^T &= e_k^T Q_{k-1,k} = s_k e_{k-1}^T - c_k e_k^T, \end{aligned}$$

so that

$$\begin{aligned} e_{k-1}^T y_k^{\mathrm{L}} &= e_{k-1}^T Q_k^T \begin{bmatrix} z_{k-1} \\ 0 \end{bmatrix} = s_{k-1} \zeta_{k-2} - c_{k-1} c_k \zeta_{k-1}, \\ e_k^T y_k^{\mathrm{L}} &= e_k^T Q_k^T \begin{bmatrix} z_{k-1} \\ 0 \end{bmatrix} = s_k \zeta_{k-1}, \\ e_k^T y_k^{\mathrm{C}} &= e_k^T Q_k^T \bar{z}_k = s_k \zeta_{k-1} - c_k \bar{\zeta}_k. \end{aligned}$$

Table 1
Storage and cost per iteration of methods based on Algorithm 2.1.

	n-vectors	dots	scal	axpy
BiLQ	6	2	3	7
BiCG	6	2	3	6
Qmr	7	2	4	7

Therefore, if we define $\mu_k = \beta_k(s_{k-1}\zeta_{k-2} - c_{k-1}c_k\zeta_{k-1}) + \alpha_k s_k \zeta_{k-1}$, $\omega_k = \beta_{k+1}s_k\zeta_{k-1}$ and $\rho_k = \beta_{k+1}(s_k\zeta_{k-1} - c_k\bar{\zeta}_k)$, we obtain

$$\|r_k^{\text{L}}\| = \sqrt{\mu_k^2\|v_k\|^2 + \omega_k^2\|v_{k+1}\|^2 + 2\mu_k\omega_k v_k^T v_{k+1}}$$

and

$$\|r_k^{\text{C}}\| = |\rho_k|\,\|v_{k+1}\|.$$

We summarize the complete procedure as Algorithm 2.2. For simplicity, we do not include a look-ahead procedure, although a robust implementation should be in order to avoid serious breakdowns (Parlett, Taylor, and Liu, 1985). The scaling factors used in our implementation are $\beta_k = |v_k^T u_k|^{\frac{1}{2}}$ and $\gamma_k = (v_k^T u_k)/\beta_k$. In Algorithm 2.2, we only form x_k^{C} at the final iteration if it is well defined. In practice, a user may choose to form it at an intermediate iteration and exit if $\bar{\delta}_k \neq 0$ and $\|r_k^{\text{C}}\|$ is sufficiently small.

Table 1 summarizes the cost per iteration of BiLQ, BiCG, and Qmr. Each method requires one operator-vector product with A and one with A^T per iteration. We assume that in-place "gemv" updates of the form $y \leftarrow Av + \gamma y$ and $y \leftarrow A^T u + \beta y$ are available. Otherwise, each method requires two additional n-vectors to store Av and $A^T u$. In the table, "dots" refers to dot products of n-vectors, "scal" refers to scaling an n-vector by a scalar, and "axpy" refers to adding a multiple of one n-vector to another one.

2.6. Properties. By construction, assuming Algorithm 2.1 does not break down, there exists an iteration $p \leq n$ such that $x_{p+1}^{\text{L}} = x_p^{\text{C}} = x_\star$, the exact solution of (1.1). In particular, there exists $y_\star$ such that $x_\star = V_p y_\star$.

Because $V_k^T U_k = I_k$ at each iteration $k \leq p$, we have

$$\|x_k^{\text{L}}\|_{U_k U_k^T} := \|U_k^T x_k^{\text{L}}\| = \|y_k^{\text{L}}\|. \tag{2.16}$$

Note that $U_k U_k^T$ is only positive semidefinite, and therefore, it does not define a norm. However, it is positive definite on Range(V_k) by biorthogonality, and $x_k^{\text{L}} \in$ Range(V_k). Thus, technically, $U_k U_k^T$ only defines a seminorm on Range(V_k). Because

$$x_k^{\text{L}} = V_k y_k^{\text{L}} = V_p \begin{bmatrix} y_k^{\text{L}} \\ 0 \end{bmatrix}, \tag{2.17}$$

we also have $\|x_k^{\text{L}}\|_{U_p U_p^T} = \|x_k^{\text{L}}\|_{U_k U_k^T}$.

The following result establishes properties of x_k^{L} that are analogous to those of the Symmlq iterate in the symmetric case.

Proposition 2.1. *Let $x_\star$ be as above. The kth* BiLQ *iterate x_k^{L} solves*

$$\underset{x}{\text{minimize}}\ \|x\|_{U_k U_k^T} \ \text{ subject to } x \in \text{Range}(V_k),\ b - Ax \perp \text{Range}(U_{k-1}) \tag{2.18}$$

Algorithm 2.2 BiLQ

```
Require: $A$, $b$, $c$
$\beta_1 v_1 = b$, $\gamma_1 u_1 = c$                                   $(\beta_1, \gamma_1)$ so that $v_1^T u_1 = 1$
$\alpha_1 = u_1^T A v_1$                                                 begin biorthogonalization
$\beta_2 v_2 = A v_1 - \alpha_1 v_1$
$\gamma_2 u_2 = A^T u_1 - \alpha_1 u_1$
$c_1 = -1$, $s_1 = 0$, $\bar{\delta}_1 = \alpha_1$                      begin LQ factorization
$\eta_1 = \beta_1$, $\bar{d}_1 = v_1$, $x_1^{\text{L}} = 0$
for $k = 2, 3, \ldots$ do
    $q = A v_k - \gamma_k v_{k-1}$, $\alpha_k = u_k^T q$                 continue biorthogonalization
    $p = A^T u_k - \beta_k u_{k-1}$
    $\beta_{k+1} v_{k+1} = q - \alpha_k v_k$                             $(\beta_{k+1}, \gamma_{k+1})$ so that $v_{k+1}^T u_{k+1} = 1$
    $\gamma_{k+1} u_{k+1} = p - \alpha_k u_k$
    $\delta_{k-1} = (\bar{\delta}_{k-1}^2 + \gamma_k^2)^{\frac{1}{2}}$    compute $Q_{k-1,k}$
    $c_k = \bar{\delta}_{k-1} / \delta_{k-1}$
    $s_k = \gamma_k / \delta_{k-1}$
    $\varepsilon_{k-2} = s_{k-1} \beta_k$                                continue LQ factorization
    $\lambda_{k-1} = -c_{k-1} c_k \beta_k + s_k \alpha_k$
    $\bar{\delta}_k = -c_{k-1} s_k \beta_k - c_k \alpha_k$
    $\zeta_{k-1} = \eta_{k-1} / \delta_{k-1}$                            update $z_{k-1}$
    $\eta_k = -\varepsilon_{k-2} \zeta_{k-2} - \lambda_{k-1} \zeta_{k-1}$
    $\mu_k = \beta_k (s_{k-1} \zeta_{k-2} - c_{k-1} c_k \zeta_{k-1}) + \alpha_k s_k \zeta_{k-1}$
    $\omega_k = \beta_{k+1} s_k \zeta_{k-1}$
    $\|r_k^{\text{L}}\| = (\mu_k^2 \|v_k\|^2 + \omega_k^2 \|v_{k+1}\|^2 + 2 \mu_k \omega_k v_k^T v_{k+1})^{\frac{1}{2}}$    compute $\|r_k^{\text{L}}\|$
    if $\bar{\delta}_k \neq 0$ then
        $\bar{\zeta}_k = \eta_k / \bar{\delta}_k$                          optional: update $\bar{z}_k$
        $\rho_k = \beta_{k+1} (s_k \zeta_{k-1} - c_k \bar{\zeta}_k)$
        $\|r_k^{\text{C}}\| = |\rho_k| \, \|v_{k+1}\|$                    optional: compute $\|r_k^{\text{C}}\|$
    end if
    $d_{k-1} = c_k \bar{d}_{k-1} + s_k v_k$                              update $\overline{D}_k$
    $\bar{d}_k = s_k \bar{d}_{k-1} - c_k v_k$
    $x_k^{\text{L}} = x_{k-1}^{\text{L}} + \zeta_{k-1} d_{k-1}$          BiLQ point
end for
if $\bar{\delta}_k \neq 0$ then
    $x_k^{\text{C}} = x_k^{\text{L}} + \bar{\zeta}_k \bar{d}_k$            optional: BiCG point
end if
```

and both

$$\underset{x}{\text{minimize}} \; \|x - x_\star\|_{U_p U_p^T} \;\; \text{subject to } x \in \text{Range}(V_p V_p^T A^T U_{k-1}), \tag{2.19a}$$

$$\underset{x}{\text{minimize}} \; \|x - x_\star\|_{U_p U_p^T} \;\; \text{subject to } x \in \text{Range}(V_k V_k^T A^T U_{k-1}). \tag{2.19b}$$

Proof. The first set of constraints of (2.18) requires that there exist $y \in \mathbb{R}^k$ such that $x = V_k y$. By biorthogonality, the objective value at such an x can be written as $\|V_k y\|_{U_k U_k^T} = \|y\|$. Biorthogonality again and (2.13) show that y_k defined in (2.3) is primal feasible for (2.18). Dual feasibility of (2.18) requires that there exist a vector q such that $y = V_k^T A^T U_{k-1} q$. By (2.1b) and biorthogonality one more time, this amounts to $y = T_{k-1,k}^T q$, which is the same as dual feasibility for (2.3). Thus, $V_k y_k^{\mathrm{L}}$ is optimal for (2.18).

To establish primal feasibility of x_k^{L} for (2.19a), note first that (2.1b) yields $A^T U_{k-1} = U_k T_{k-1,k}^T$. Let $\bar{V}_{p-k}$ denote the last $p-k$ columns of V_p. Biorthogonality yields

$$V_p^T U_k = \begin{bmatrix} V_k^T \\ \bar{V}_{p-k}^T \end{bmatrix} U_k = \begin{bmatrix} I_k \\ 0 \end{bmatrix} \quad \text{and} \quad V_p V_p^T U_k = V_k.$$

As in the first part of the proof, $y_k^{\mathrm{L}} = T_{k-1,k}^T q$ for some $q \in \mathbb{R}^{k-1}$, and therefore, $x_k^{\mathrm{L}} = V_p V_p^T A^T U_{k-1} q$. Dual feasibility requires that

$$\begin{aligned} 0 &= U_{k-1}^T A V_p V_p^T U_p U_p^T (x_k^{\mathrm{L}} - x_\star) \\ &= U_{k-1}^T A V_p U_p^T V_p \left(\begin{bmatrix} y_k^{\mathrm{L}} \\ 0 \end{bmatrix} - y_\star \right) \\ &= U_{k-1}^T A (x_k^{\mathrm{L}} - x_\star) \\ &= -U_{k-1}^T r_k^{\mathrm{L}}, \end{aligned}$$

where we used biorthogonality and (2.17) and is satisfied because of (2.13).

Finally, x_k^{L} also solves (2.19b) because $V_p V_p^T A^T U_{k-1} = V_k V_k^T A^T U_{k-1}$. □

Note that (2.18) continues to hold if the objective is measured in the $(U_p U_p^T)$-seminorm. Although this seminorm is no longer iteration dependent, it is unknown until the end of the biorthogonalization process.

Because $\mathrm{Range}(V_p V_p^T A^T U_{k-1}) \subseteq \mathrm{Range}(V_p V_p^T A^T U_k)$ (2.19a) indicates that

$$\|x_{k+1}^{\mathrm{L}} - x_\star\|_{U_p U_p^T} \le \|x_k^{\mathrm{L}} - x_\star\|_{U_p U_p^T};$$

i.e., the $(U_p U_p^T)$-seminorm of the error is monotonic. The identity (2.17) and biorthogonality then imply that

$$\|x_k^{\mathrm{L}} - x_\star\|_{U_p U_p^T} = \left\| V_p \left(\begin{bmatrix} y_k^{\mathrm{L}} \\ 0 \end{bmatrix} - y_\star \right) \right\|_{U_p U_p^T} = \left\| \begin{bmatrix} y_k^{\mathrm{L}} \\ 0 \end{bmatrix} - y_\star \right\| \tag{2.20}$$

is also monotonically decreasing.

Note that (2.16) is readily computable because $\|y_k^{\mathrm{L}}\| = \|z_{k-1}\|$, where z_{k-1} is defined at the beginning of section 2.4 and can be updated as

$$\|x_{k+1}^{\mathrm{L}}\|_{U_{k+1} U_{k+1}^T}^2 = \|x_k^{\mathrm{L}}\|_{U_k U_k^T}^2 + \zeta_k^2. \tag{2.21}$$

It is apparent from (2.21) that the $(U_k U_k^T)$- and $(U_p U_p^T)$-seminorms of the BiLQ iterates are monotonically increasing. The identity (2.17) and biorthogonality again imply that the same goes with $\|y_k^{\mathrm{L}}\|$.

A lower bound on the error (2.20) can be obtained as $\|z_{k-d} - z_{k-1}\|$ for a user-defined *delay* of d iterations. Such a lower bound may be used to define a simple though not robust error-based stopping criterion (Estrin, Orban, and Saunders, 2019b).

In the symmetric case, where $V_k = U_k$ is orthogonal and $T_k = T_k^T$, the Symmlq iterate solves the problem

$$\underset{x}{\text{minimize}} \; \|x - x_\star\| \;\text{ subject to } x \in \mathrm{Range}(A V_{k-1}), \tag{2.22}$$

which coincides with (2.19).

We now establish a similar property for x_k^{C} provided that it is well defined.

PROPOSITION 2.2. *Let $x_\star$ be as above. If T_k is nonsingular, the kth* BICG *iterate x_k^{C} is well defined and solves*

$$\underset{x}{\text{minimize}} \ \|x - x_\star\|_{U_p U_p^T} \ \text{ subject to } x \in \text{Range}(V_k V_k^T A^T U_k). \tag{2.23}$$

Proof. Primal feasibility for (2.23) follows from the fact that $V_k^T A^T U_k = T_k^T$ and $x_k^{\mathrm{C}} = V_k y_k^{\mathrm{C}} = V_k T_k^T q$. Dual feasibility follows from (2.14) as in the proof of Proposition 2.1. □

Contrary to Proposition 2.1, there is no second problem involving V_p in (2.23) because $V_p V_p^T A^T U_k = V_{k+1} T_{k,k+1}^T$, whereas $V_k V_k^T A^T U_k = T_k^T$.

2.7. Numerical experiments. Nonhomogeneous linear PDEs with variable coefficients of the form

$$\sum_{i=1}^{n} \sum_{j=1}^{p} a_{i,j}(x) \frac{\partial^j u(x)}{\partial x_i^j} = b(x) \tag{2.24}$$

are frequent when physical phenomena are modeled in polar, cylindrical, or spherical coordinates. The discretization of (2.24) often leads to a nonsymmetric square system. Such is the case with Poisson's equation $\Delta u = f$, used, for instance, to describe the gravitational or electrostatic field caused by a given mass density or charge distribution. The two-dimensional Poisson equation in polar coordinates with Dirichlet boundary conditions is

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r \frac{\partial u(r,\theta)}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 u(r,\theta)}{\partial \theta^2} = f(r,\theta), \quad (r,\theta) \in (0,R) \times [0, 2\pi), \tag{2.25a}$$

$$u(R,\theta) = g(\theta), \qquad \theta \in [0, 2\pi), \tag{2.25b}$$

where $R > 0$, the source term f, and the boundary condition g are given. We discretize (2.25) using central differences with 50 discretization points for r and 50 for θ, with $g(\theta) = 0$, $f(r,\theta) = -3\cos(\theta)$, and $R = 1$ so that (2.25) models the response of an attached circular elastic membrane to a force. The resulting matrix has size 2,500 with 12,400 nonzeros and is block tridiagonal with extra diagonal blocks in the northeast and southwest corners. Each block on the main diagonal is tridiagonal but not symmetric. Each off-diagonal block is diagonal. More details on the discretization used are given by Lai (2001). The exact solution is represented in Figure 1.

We compare BILQ with our implementation of QMR without look-ahead. We also simulate BICG by way of the transition from x_k^{L} to x_k^{C} in Algorithm 2.2. Figure 2 reports the residual and error history of BILQ, BICG, and QMR on (2.25). To compute $\|r_k\|$ and $\|e_k\|$, residuals $b - Ax_k$ and errors $x_k - x_\star$ are explicitly calculated at each iteration. We compute a reference solution with Julia's backslash command. We run each method with an absolute tolerance $\varepsilon_a = 10^{-10}$ and a relative tolerance $\varepsilon_r = 10^{-7}$ such that the algorithms stop when $\|r_k\| \le \varepsilon_a + \|b\|\varepsilon_r$.

We also compare BILQ with BICG and QMR on the matrices SHERMAN5 and RAEFSKY1, with their respective right-hand side, from the UFL collection of Davis and Hu (2011).[1] System SHERMAN5 has size 3,312 with 20,793 nonzeros, and RAEFSKY1 has size 3,242 with 293,409 nonzeros. A diagonal preconditioner is used for both systems.

[1] Now the SuiteSparse Matrix Collection, https://sparse.tamu.edu.

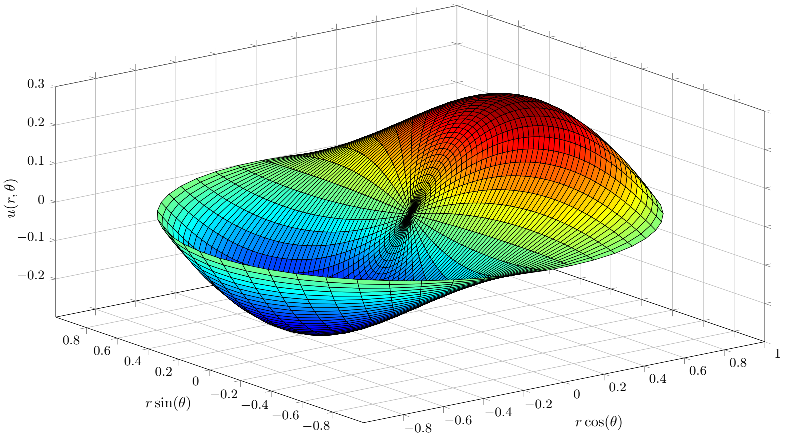

FIG. 1. *Solution* $u(r,\theta) = r(1-r)\cos(\theta)$ *of* (2.25) *with* $g(\theta) = 0$, $f(r,\theta) = -3\cos(\theta)$ *and* $R = 1$.

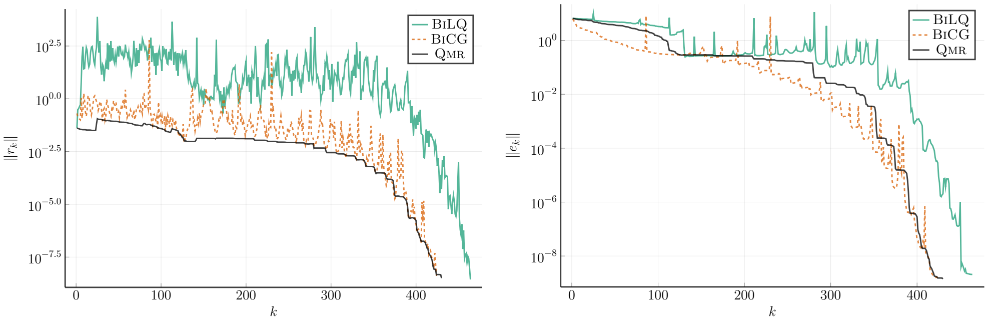

FIG. 2. *Convergence curves of* BiLQ, BiCG, *and* Qmr *iterates on* (2.25). *The figures show the residual (left) and error (right) history for each method.*

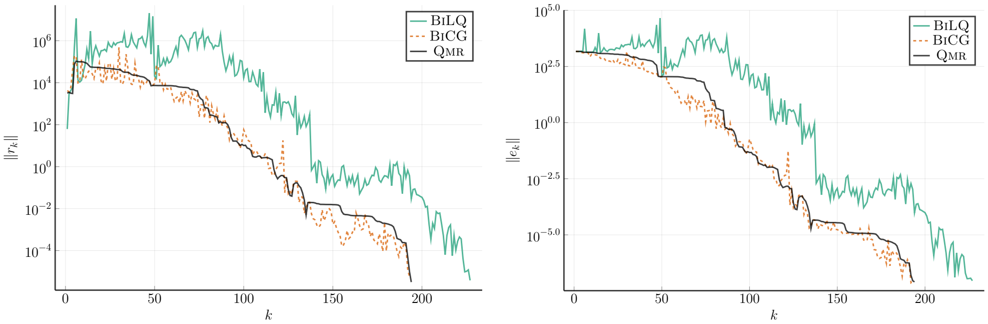

FIG. 3. *Convergence curves of* BiLQ, BiCG, *and* Qmr *iterates for the SHERMAN*5 *system. The figures show the residual (left) and error (right) history for each method.*

Figures 2–4 all show that in BiLQ, neither the residual nor the error are monotonic in general. They also appear more erratic than Qmr. As in the symmetric case, both generally lag compared to those of BiCG and Qmr but are not far behind. We experimented with other systems and observed the same qualitative behavior. As shown in section 2.6, although BiLQ is a minimum-error–type method, this error is

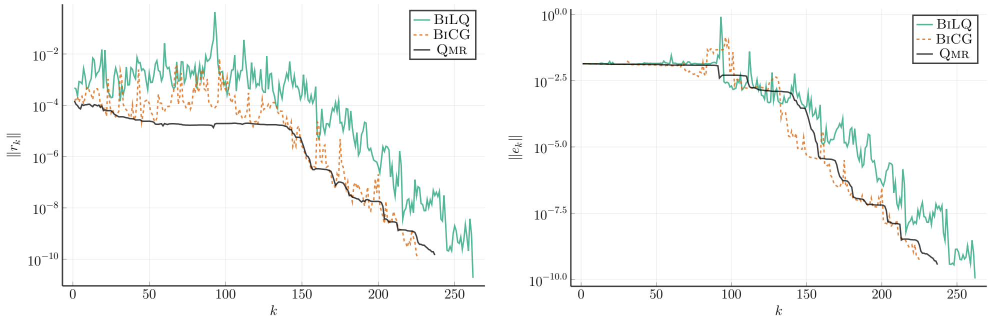

Fig. 4. *Convergence curves of* BiLQ, BiCG, *and* Qmr *iterates for the RAEFSKY1 system. The figures show the residual (left) and error (right) history for each method.*

minimized over a different space than that where x_k^{L} and x_k^{C} reside; see Proposition 2.1. This situation is analogous to that between Symmlq and Cg in the symmetric case (Estrin, Orban, and Saunders, 2019c). Thus, the possibility of transferring to the BiCG point, when it exists, is attractive. Because the BiCG residual is easily computable, transferring based on the residual norm is readily implemented. The determination of upper bounds on the error suitable as stopping criteria remains the subject of active research (Estrin, Orban, and Saunders, 2019a,b,c).

2.8. Discussion. Like Qmr, the BiLQ iterate is well defined at each step even if T_k is singular, whereas x_k^{C} is undefined when $\bar{\delta}_k = 0$. A simple example is

$$A = \begin{bmatrix} 0 & -1 \\ 1 & 1 \end{bmatrix}, \qquad b = c = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

According to Algorithm 2.1, $\beta_1 = \gamma_1 = 1$, $v_1 = u_1 = b = c$. Then $\alpha_1 = u_1^T A v_1 = 0$, $T_1 = \begin{bmatrix}\alpha_1\end{bmatrix}$ is singular, and $T_1 y_1 = \beta_1$ is inconsistent. BiCG and its variants Cgs (Sonneveld, 1989) and BiCGStab (van der Vorst, 1992) all fail. However, T_2 is not singular, and the BiCG point exists, although we cannot compute it without look-ahead. In finite-precision arithmetic, such exact breakdowns are rather rare. But near breakdowns ($\bar{\delta}_k \approx 0$) may happen and lead to numerical instabilities in ensuing iterations. Another drawback of BiCG is that the LU decomposition of T_k might not exist without pivoting even if T_k is nonsingular, whereas (2.6) is always well defined.

3. Adjoint systems.

Motivated by fluid dynamics applications, Pierce and Giles (2000) describe a method for doubling the order of accuracy of estimates of integral functionals involving the solution of a PDE. Consider a well-posed linear PDE $Lu = f$ on a domain Ω subject to homogeneous boundary conditions, where L is a differential operator of the form (2.24) and $f \in L_2(\Omega)$. Suppose we wish to evaluate the functional $J(u) := \langle u,\, g\rangle$, where $g \in L_2(\Omega)$ and $\langle \cdot,\, \cdot\rangle$ represents an integral inner product on $L_2(\Omega)$. The problem may be stated equivalently as evaluating the functional $\langle v,\, f\rangle$, where v solves the adjoint PDE $L^* v = g$ because $\langle v,\, f\rangle = \langle v,\, Lu\rangle = \langle L^* v,\, u\rangle = \langle g,\, u\rangle$.

Let the discretization of L yield the linear system $Au_D = f_D$ with D a set of points that define a grid on Ω. For certain types of PDEs and certain discretization schemes, A^T is an appropriate discretization of L^*. Pierce and Giles (2000) provide examples with linear operators such as Poisson's equation discretized by finite differences in 1D and by finite elements in 2D, but their discretizations are symmetric. Their method also applies to cases where $A \neq A^T$, but in such cases, the discretizations of the primal and dual equations commonly differ. Therefore, there is a need for methods that solve an unsymmetric primal system and its adjoint simultaneously. Lu and Darmofal (2003) and Golub, Stoll, and Wathen (2008) were also interested in this problem for scattering amplitude evaluation. Lu and Darmofal (2003) devise a modification of Qmr in which the two initial vectors are b and c and a quasi residual is minimized for both the primal and adjoint systems via an updated QR factorization. Golub, Stoll, and Wathen (2008) apply Usymqr (Saunders, Simon, and Yip, 1988) to both the primal and the adjoint system[2] simultaneously by updating two QR factorizations. The advantage of their approach is that it produces monotonic residuals for both systems.

Assume we use a method to compute u_D and to solve $A^T v_D = g_D$ such that $\|u - u_D\| \in \mathcal{O}(h^p)$ and $\|v - v_D\| \in \mathcal{O}(h^p)$, where h describes the grid coarseness. From u_D and v_D, we compute approximations $u_h \approx u$ and $v_h \approx v$ over Ω by way of an interpolation of higher order than the discretization. Define $f_h := Lu_h$ and $g_h := L^* v_h$. Instead of $J(u) \approx \langle u_h, g \rangle$, an approximation of order p, we may obtain one of order $2p$ via the identity

$$\langle g, u \rangle = \langle g, u_h \rangle - \langle v_h, f_h - f \rangle + \langle g_h - g, u_h - u \rangle. \tag{3.1}$$

The first two terms constitute our new approximation, while the remaining error term can be expressed as $\langle g_h - g, L^{-1}(f_h - f) \rangle = \mathcal{O}(h^{2p})$.

From this point, we consider, in addition to (1.1), the adjoint system

$$A^T t = c. \tag{3.2}$$

Solving simultaneously primal and dual systems can also be formulated as solving the symmetric and indefinite system

$$\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} t \\ x \end{bmatrix} = \begin{bmatrix} b \\ c \end{bmatrix}. \tag{3.3}$$

Minres or Minres-qlp (Choi, Paige, and Saunders, 2011) are prime candidates for (3.3) and will serve as a basis for comparison. We use Minres-qlp in the numerical experiments below, as it should be more stable numerically, but Minres should behave similarly because the two methods generate the same iterates in exact arithmetic on consistent systems. For this reason, we refer to Minres and Minres-qlp somewhat interchangeably in the following.

In the context of Algorithm 2.1, we can take advantage of the two initial vectors b and c to combine BiLQ and Qmr and solve both the primal and adjoint systems simultaneously at no other extra cost than that of updating solution and residual estimates. We call the resulting method BiLQR. Contrary to the approach of Lu and Darmofal (2003), no extra factorization updates are necessary. Instead of approximating u_D and v_D by minimizing two quasi residuals, BiLQR minimizes one quasi residual and computes the second approximation via a minimum-norm subproblem.

A similar method based on the orthogonal tridiagonalization process of Saunders, Simon, and Yip (1988) can be derived by combining Usymlq and Usymqr, which we call TriLQR and which is to the approach of Golub, Stoll, and Wathen (2008) as BiLQR is to that of Lu and Darmofal (2003). TriLQR remains well defined for rectangular A (Reichel and Ye, 2008).

[2] Although they follow Reichel and Ye (2008) and call Usymqr the "generalized Lsqr."

3.1. Description of BiLQR. BiLQR updates an approximate solution $t^{\mathrm{Q}}_{k-1} = U_{k-1} f^{\mathrm{Q}}_{k-1}$ of $A^T t = c$ by solving the Qmr least-squares subproblem

$$\underset{f}{\text{minimize}}\ \|T^T_{k-1,k} f - \gamma_1 e_1\| \quad \Longleftrightarrow \quad \underset{f}{\text{minimize}}\ \left\| \begin{bmatrix} L^T_{k-1} \\ 0 \end{bmatrix} f - Q_k \gamma_1 e_1 \right\| \tag{3.4}$$

because the QR factorization of $T^T_{k-1,k}$ is readily available. Define $\bar{h}_k = Q_k \gamma_1 e_1 = (h_{k-1},\, \bar{\psi}_k) = (\psi_1,\, \ldots,\, \psi_{k-1},\, \bar{\psi}_k)$. The components of $\bar{h}_k$ are updated according to

$$\bar{\psi}_1 = \gamma_1, \tag{3.5a}$$
$$\psi_k = c_{k+1} \bar{\psi}_k, \quad k \geq 1, \tag{3.5b}$$
$$\bar{\psi}_{k+1} = s_{k+1} \bar{\psi}_k, \quad k \geq 1. \tag{3.5c}$$

The solution of (3.4) is $f^{\mathrm{Q}}_{k-1} = L^{-T}_{k-1} h_{k-1}$, and the least-squares residual norm is $|\bar{\psi}_k|$. To avoid storing U_k, we define $W_k = U_k L_k^{-T}$, which can be updated as

$$w_1 = u_1/\delta_1, \tag{3.6a}$$
$$w_2 = (u_2 - \lambda_1 w_1)/\delta_2, \tag{3.6b}$$
$$w_k = (u_k - \lambda_{k-1} w_{k-1} - \varepsilon_{k-2} w_{k-2})/\delta_k, \quad k \geq 3. \tag{3.6c}$$

At the next iteration, t^{Q}_k can be recursively updated according to

$$t^{\mathrm{Q}}_k = U_k f^{\mathrm{Q}}_k = U_k L_k^{-T} h_k = W_k h_k = W_{k-1} h_{k-1} + \psi_k w_k = t^{\mathrm{Q}}_{k-1} + \psi_k w_k.$$

The Qmr residual is

$$r^{\mathrm{Q}}_k = c - A^T t^{\mathrm{Q}}_k = U_{k+1}(\gamma_1 e_1 - T^T_{k,k+1} f^{\mathrm{Q}}_k) = \bar{\psi}_{k+1} U_{k+1} Q^T_{k+1} e_{k+1},$$

so that

$$\|r^{\mathrm{Q}}_k\| \leq \|U_{k+1}\|_F \, \|\bar{\psi}_{k+1} Q^T_{k+1} e_{k+1}\| \leq |\bar{\psi}_{k+1}| \sqrt{\tau_{k+1}},$$

where $\tau_{k+1} = \sum_{i=1}^{k+1} \|u_i\|^2 = \tau_k + \|u_{k+1}\|^2$. If the u_k are normalized, then $\tau_k = k$. Algorithm 3.2 states the complete procedure.

The following result states a minimization property of the Qmr residual in an iteration-dependent norm.

Proposition 3.1. *The* $(k-1)$*th* Qmr *iterate* t^{Q}_{k-1} *solves*

$$\underset{t}{\text{minimize}}\ \|c - A^T t\|_{V_k V_k^T} \ \text{ subject to } t \in \text{Range}(U_{k-1}). \tag{3.7}$$

In addition, $\|r^{\mathrm{Q}}_k\|_{V_k V_k^T}$ *is monotonically decreasing.*

Proof. The set of constraints of (3.7) requires that there exist $f \in \mathbb{R}^{k-1}$ such that $t = U_{k-1} f$. By biorthogonality, the objective value at such a t can be written as $\|c - A^T U_{k-1} f\|_{V_k V_k^T} = \|c - U_k T^T_{k-1,k} f\|_{V_k V_k^T} = \|\gamma_1 e_1 - T^T_{k-1,k} f\|$. We recover the subproblem (3.4).

For the second part, $\|r^{\mathrm{Q}}_k\|_{V_{k+1} V^T_{k+1}} = |\bar{\psi}_{k+1}| = |s_{k+1}| |\bar{\psi}_k| = |s_{k+1}| \|r^{\mathrm{Q}}_{k-1}\|_{V_k V_k^T}$. □

Note that Proposition 3.1 continues to hold if r^{Q}_k is measured in the $V_p V_p^T$-norm.

Algorithm 3.1 Orthogonal Tridiagonalization Process.

```
Require: $A$, $b$, $c$
$v_0 = 0$, $u_0 = 0$
$\beta_1 v_1 = b$, $\gamma_1 u_1 = c$                              $(\beta_1,\ \gamma_1) > 0$ so that $\|v_1\| = \|u_1\| = 1$
for $k = 1, 2, \dots$ do
    $q = Au_k - \gamma_k v_{k-1}$, $\alpha_k = v_k^T q$
    $p = A^T v_k - \beta_k u_{k-1}$
    $\beta_{k+1} v_{k+1} = q - \alpha_k v_k$                       $\beta_{k+1} > 0$ so that $\|v_{k+1}\| = 1$
    $\gamma_{k+1} u_{k+1} = p - \alpha_k u_k$                      $\gamma_{k+1} > 0$ so that $\|u_{k+1}\| = 1$
end for
```

3.2. Description of TriLQR.

The Saunders, Simon, and Yip (1988) tridiagonalization process generates sequences of vectors $\{v_k\}$ and $\{u_k\}$ such that $v_i^T v_j = \delta_{ij}$ and $u_i^T u_j = \delta_{ij}$ in exact arithmetic. The process is summarized as Algorithm 3.1.

At the end of the kth iteration, we have

$$AU_k = V_k T_k \ + \beta_{k+1} v_{k+1} e_k^T = V_{k+1} T_{k+1,k}, \tag{3.8a}$$

$$A^T V_k = U_k T_k^T + \gamma_{k+1} u_{k+1} e_k^T = U_{k+1} T_{k,k+1}^T, \tag{3.8b}$$

to be compared with (2.1).

Saunders, Simon, and Yip (1988) develop two methods based on Algorithm 3.1. USYMLQ generates an approximation to a solution of (1.1) of the form $x_k^{\text{LQ}} = U_k y_k^{\text{LQ}}$, where $y_k^{\text{LQ}} \in \mathbb{R}^k$ solves

$$\underset{y}{\text{minimize}} \ \|y\| \quad \text{subject to} \ T_{k-1,k} y = \beta_1 e_1. \tag{3.9}$$

With (3.8) and (3.9), we have the following analogue of Proposition 2.1 and (2.22).

PROPOSITION 3.2. *Let $x_\star$ be the exact solution of* (1.1). *The kth* USYMLQ *iterate* x_k^{LQ} *solves*

$$\underset{x}{\text{minimize}} \ \|x\| \ \text{subject to} \ x \in \text{Range}(U_k), \ b - Ax \perp \text{Range}(U_{k-1}) \tag{3.10}$$

and

$$\underset{x}{\text{minimize}} \ \|x - x_\star\| \ \text{subject to} \ x \in \text{Range}(A^T V_{k-1}). \tag{3.11}$$

Proof. The proof is nearly identical to that of Proposition 2.1 and relies on the fact that $r_k^{\text{LQ}} := b - Ax_k^{\text{LQ}}$ is a combination of u_k and u_{k+1} (Buttari et al., 2019, section 3.2.2). □

Similar to BiCG, there exists a method based on Algorithm 3.1 that one might name USYMCG. USYMCG generates an approximation of the form $x_k^{\text{CG}} = U_k y_k^{\text{CG}}$, where $y_k^{\text{CG}} \in \mathbb{R}^k$ solves

$$T_k y = \beta_1 e_1. \tag{3.12}$$

It shares the main disadvantage of BiCG; that is, it fails to define an iterate when T_k is singular. However, when the iterate exists, a transfer from the USYMLQ point is possible. This method is also alluded to by Buttari et al. (2019).

Table 2
Storage and cost per iteration of methods for solving (1.1) *and* (3.2) *simultaneously.*

	n-vectors	dots	scal	axpy
BiLQR	9	2	5	10
TriLQR	9	2	5	10
Minres-qlp	10	4	8	14
Lu and Darmofal (2003)	10	2	6	10
Golub, Stoll, and Wathen (2008)	10	2	6	10

The second method, Usymqr, used to solve (3.2) generates an approximation $t_k^{\text{QR}} = V_k f_k^{\text{QR}}$, where $f_k^{\text{QR}} \in \mathbb{R}^k$ solves

$$\underset{f}{\text{minimize}} \ \|T_{k,k+1}^T f - \gamma_1 e_1\|. \tag{3.13}$$

The following property applies to t_k^{QR} due to our assumption that (1.1) is consistent.

Proposition 3.3 (Buttari et al., 2019, Theorem 1). *Assume* $b \in \text{Range}(A)$. *Then* Usymqr *finds the minimum-norm solution of*

$$\underset{t}{\text{minimize}} \ \|A^T t - c\|.$$

Of course, A nonsingular implies that the solution to (3.2) is unique, but Proposition 3.3 applies more generally to rectangular and/or rank-deficient A.

When $A = A^T$ and $b = c$, Algorithm 3.1 coincides with the symmetric Lanczos process, and Usymlq and Usymqr are equivalent to Symmlq and Minres (Paige and Saunders, 1975), respectively. Besides the orthogonalization process, differences between those methods and BiLQ, BiCG, and Qmr are the definition of $\bar{D}_k$ and W_k and the fact that u_k and v_k are swapped. If stopping criteria are based on residual norms, expressions derived for methods based on Algorithm 2.1 apply to methods based on Algorithm 3.1, but their expressions can be simplified because V_k and U_k are both orthogonal:

$$\|r_k^{\text{LQ}}\| = \sqrt{\mu_k^2 + \omega_k^2}, \quad \|r_k^{\text{CG}}\| = |\rho_k|, \quad \text{and} \quad \|r_k^{\text{QR}}\| = |\bar{\psi}_{k+1}|.$$

Usymlq and Usymqr can be combined into TriLQR to solve both the primal and adjoint system simultaneously. We summarize the complete procedure as Algorithm 3.3 and highlight lines with differences between the two algorithms.

BiLQR and TriLQR both need nine n-vectors: u_k, u_{k-1}, v_k, v_{k-1}, w_k, w_{k-1}, $\bar{d}_k$, x_k, and t_{k-1}, whereas Minres-qlp applied to (3.3) can be implemented with five $(2n)$-vectors. Two more n-vectors are needed when in-place "gemv" updates are not explicitly available. Table 2 summarizes the cost of BiLQR, TriLQR, Minres-qlp, and variants from Lu and Darmofal (2003) and Golub, Stoll, and Wathen (2008), developed for adjoint systems. An advantage of Minres-qlp and TriLQR is that adjoint systems can be solved even if $b^T c = 0$, which is not possible with BiLQR. In addition, breakdowns $q^T p = 0$ with $p \neq 0$ and $q \neq 0$ are not a problem with TriLQR. TriLQR is similar in spirit to the recent method Usymlqr of Buttari et al. (2019) for solving symmetric saddle-point systems but is slightly cheaper.

3.3. Applications.

As a simple illustration, we consider a 1D ODE and a 2D PDE. Consider first the linear ODE with constant coefficients

$$\chi_1 u''(x) + \chi_2 u'(x) + \chi_3 u(x) = f(x) \quad x \in \Omega, \tag{3.14a}$$

$$u(x) = 0 \quad x \in \partial\Omega, \tag{3.14b}$$

Algorithm 3.2 BiLQR

```
Require: $A$, $b$, $c$
$\beta_1 v_1 = b$, $\gamma_1 u_1 = c$
$\alpha_1 = u_1^T A v_1$
$\beta_2 v_2 = A v_1 - \alpha_1 v_1$
$\gamma_2 u_2 = A^T u_1 - \alpha_1 u_1$
$c_1 = -1$, $s_1 = 0$, $\bar{\delta}_1 = \alpha_1$
$\eta_1 = \beta_1$, $\bar{d}_1 = v_1$, $\bar{\psi}_1 = \gamma_1$
$x_1^{\mathrm{L}} = 0$, $t_0^{\mathrm{Q}} = 0$
for $k = 2, 3, \ldots$ do
    $q = A v_k - \gamma_k v_{k-1}$, $\alpha_k = u_k^T q$
    $p = A^T u_k - \beta_k u_{k-1}$
    $\beta_{k+1} v_{k+1} = q - \alpha_k v_k$
    $\gamma_{k+1} u_{k+1} = p - \alpha_k u_k$
    $\delta_{k-1} = (\bar{\delta}_{k-1}^2 + \gamma_k^2)^{\frac{1}{2}}$
    $c_k = \bar{\delta}_{k-1} / \delta_{k-1}$
    $s_k = \gamma_k / \delta_{k-1}$
    $\varepsilon_{k-2} = s_{k-1} \beta_k$
    $\lambda_{k-1} = -c_{k-1} c_k \beta_k + s_k \alpha_k$
    $\bar{\delta}_k = -c_{k-1} s_k \beta_k - c_k \alpha_k$
    $\zeta_{k-1} = \eta_{k-1} / \delta_{k-1}$
    $\eta_k = -\varepsilon_{k-2} \zeta_{k-2} - \lambda_{k-1} \zeta_{k-1}$
    $d_{k-1} = c_k \bar{d}_{k-1} + s_k v_k$
    $\bar{d}_k = s_k \bar{d}_{k-1} - c_k v_k$
    $\psi_{k-1} = c_k \bar{\psi}_{k-1}$
    $\bar{\psi}_k = s_k \bar{\psi}_{k-1}$
    $w_{k-1} = \frac{u_{k-1} - \lambda_{k-2} w_{k-2} - \varepsilon_{k-3} w_{k-3}}{\delta_{k-1}}$
    $x_k^{\mathrm{L}} = x_{k-1}^{\mathrm{L}} + \zeta_{k-1} d_{k-1}$
    $t_{k-1}^{\mathrm{Q}} = t_{k-2}^{\mathrm{Q}} + \psi_{k-1} w_{k-1}$
end for
if $\bar{\delta}_k \neq 0$ then
    $\bar{\zeta}_k = \eta_k / \bar{\delta}_k$
    $x_k^{\mathrm{C}} = x_k^{\mathrm{L}} + \bar{\zeta}_k \bar{d}_k$
end if
```

Algorithm 3.3 TriLQR

```
Require: $A$, $b$, $c$
$\beta_1 v_1 = b$, $\gamma_1 u_1 = c$
$\alpha_1 = u_1^T A v_1$
$\beta_2 v_2 = A u_1 - \alpha_1 v_1$
$\gamma_2 u_2 = A^T v_1 - \alpha_1 u_1$
$c_1 = -1$, $s_1 = 0$, $\bar{\delta}_1 = \alpha_1$
$\eta_1 = \beta_1$, $\bar{d}_1 = u_1$, $\bar{\psi}_1 = \gamma_1$
$x_1^{\mathrm{LQ}} = 0$, $t_0^{\mathrm{QR}} = 0$
for $k = 2, 3, \ldots$ do
    $q = A u_k - \gamma_k v_{k-1}$, $\alpha_k = v_k^T q$
    $p = A^T v_k - \beta_k u_{k-1}$
    $\beta_{k+1} v_{k+1} = q - \alpha_k v_k$
    $\gamma_{k+1} u_{k+1} = p - \alpha_k u_k$
    $\delta_{k-1} = (\bar{\delta}_{k-1}^2 + \gamma_k^2)^{\frac{1}{2}}$
    $c_k = \bar{\delta}_{k-1} / \delta_{k-1}$
    $s_k = \gamma_k / \delta_{k-1}$
    $\varepsilon_{k-2} = s_{k-1} \beta_k$
    $\lambda_{k-1} = -c_{k-1} c_k \beta_k + s_k \alpha_k$
    $\bar{\delta}_k = -c_{k-1} s_k \beta_k - c_k \alpha_k$
    $\zeta_{k-1} = \eta_{k-1} / \delta_{k-1}$
    $\eta_k = -\varepsilon_{k-2} \zeta_{k-2} - \lambda_{k-1} \zeta_{k-1}$
    $d_{k-1} = c_k \bar{d}_{k-1} + s_k u_k$
    $\bar{d}_k = s_k \bar{d}_{k-1} - c_k u_k$
    $\psi_{k-1} = c_k \bar{\psi}_{k-1}$
    $\bar{\psi}_k = s_k \bar{\psi}_{k-1}$
    $w_{k-1} = \frac{v_{k-1} - \lambda_{k-2} w_{k-2} - \varepsilon_{k-3} w_{k-3}}{\delta_{k-1}}$
    $x_k^{\mathrm{LQ}} = x_{k-1}^{\mathrm{LQ}} + \zeta_{k-1} d_{k-1}$
    $t_{k-1}^{\mathrm{QR}} = t_{k-2}^{\mathrm{QR}} + \psi_{k-1} w_{k-1}$
end for
if $\bar{\delta}_k \neq 0$ then
    $\bar{\zeta}_k = \eta_k / \bar{\delta}_k$
    $x_k^{\mathrm{CG}} = x_k^{\mathrm{LQ}} + \bar{\zeta}_k \bar{d}_k$
end if
```

where $\Omega = [0, 1]$, and say we are interested in the value of the linear functional

$$J(u) = \int_{\Omega} u(x) g(x) \, \mathrm{d}x, \tag{3.15}$$

where u solves (3.14) and $g \in L_2(\Omega)$. The adjoint equation can be derived from (3.14) using integration by parts:

$$\chi_1 v''(x) - \chi_2 v'(x) + \chi_3 v(x) = g(x) \quad x \in \Omega, \tag{3.16a}$$

$$v(x) = 0 \quad x \in \partial\Omega. \tag{3.16b}$$

Note that the only difference between the primal and adjoint equations resides in the sign of odd-degree derivatives. The discussion in section 3 ensures that

$$G(v) := \int_\Omega f(x)v(x)\,\mathrm{d}x = J(u). \tag{3.17}$$

Consider the uniform discretization $x_i = ih$, $i = 0, \dots, N+1$, where $h = 1/(N+1)$. We use central differences of order 2, i.e.,

$$u'(x_i) = \frac{u_{i+1} - u_{i-1}}{2h} + \mathcal{O}(h^2), \quad u''(x_i) = \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} + \mathcal{O}(h^2).$$

We obtain $u(x_i)$ for $x_i \in D := \{x_i \mid i \in 1, \dots, N\}$ from the tridiagonal linear system

$$\begin{bmatrix} -2\chi_1 + \chi_3 h^2 & \chi_1 + \chi_2 h & & \\ \chi_1 - \chi_2 h & -2\chi_1 + \chi_3 h^2 & \ddots & \\ & \ddots & \ddots & \chi_1 + \chi_2 h \\ & & \chi_1 - \chi_2 h & -2\chi_1 + \chi_3 h^2 \end{bmatrix} \begin{bmatrix} u(x_1) \\ \vdots \\ \vdots \\ u(x_N) \end{bmatrix} = h^2 \begin{bmatrix} f(x_1) \\ \vdots \\ \vdots \\ f(x_N) \end{bmatrix}.$$

More compactly, we write $Au_D = f_D$. Similarly, we compute $v(x_i)$ for $x_i \in D$ from $A^T v_D = g_D$. Next, we compute an approximation of u and v over Ω by cubic spline interpolation, and the resulting functions are denoted u_h and v_h. We require that $Lu_h = f$ and $L^* v_h = g$ on $\partial\Omega$. We subsequently obtain $f_h(x) := \chi_1 u_h''(x) + \chi_2 u_h'(x) + \chi_3 u_h(x)$. The end-point conditions of the cubic splines require that f_h coincide with f on $\partial\Omega$. Finally, we compute the improved estimate (3.1) using a three-point Gauss quadrature to approximate each

$$\int_{x_i}^{x_{i+1}} g(x)u_h(x)\,\mathrm{d}x - \int_{x_i}^{x_{i+1}} v_h(x)(f_h(x) - f(x))\,\mathrm{d}x$$

on each subinterval to ensure that the numerical quadrature errors are smaller than the discretization error.

We choose $n = 50$, $\chi_1 = \chi_2 = \chi_3 = 1$, $g(x) = e^x$, and $f(x)$ such that the exact solution of (3.14) is $u_\star(x) = \sin(\pi x)$. The resulting linear system has dimension 50 with 148 nonzeros. Those parameters ensure that $J_\star = \langle g, u_\star \rangle = (\pi(e+1))/(\pi^2+1)$. Figures 5 and 6 report the evolution of the residual and error on (1.1) and (3.2) for (3.14) and (3.16), respectively. BiLQR terminates in 51 iterations, TriLQR in 87 iterations, and Minres-qlp in 198 iterations. The left plot of Figure 7 illustrates the error in the evaluation of $J(u)$ as a function of h using the naive $J(u) \approx J(u_h)$ and improved (3.1) approximations.

The steady-state convection-diffusion equation with constant coefficients

$$\kappa_1 \Delta u(x) + \kappa_2 \nabla \cdot u(x) = f(x) \quad x \in \Omega, \tag{3.18a}$$

$$u(x) = 0 \qquad x \in \partial\Omega, \tag{3.18b}$$

where $f \in L_2(\Omega)$, describes the flow of heat, particles, or other physical quantities in situations where there is both diffusion and convection or advection. Assume as before

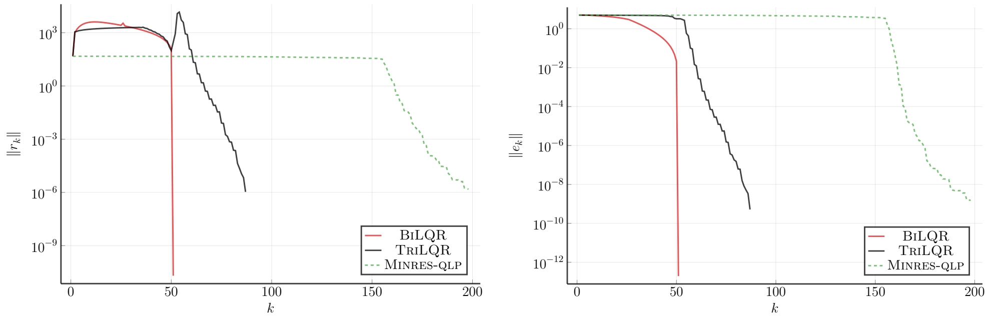

FIG. 5. *Residuals and errors norms of* BiLQR, TriLQR, *and* Minres-qlp *iterates on* (3.14).

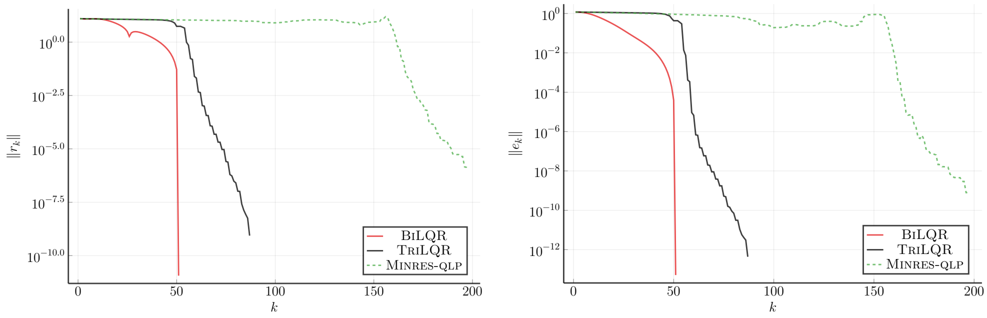

FIG. 6. *Residuals and errors norms of* BiLQR, TriLQR, *and* Minres-qlp *iterates on* (3.16).

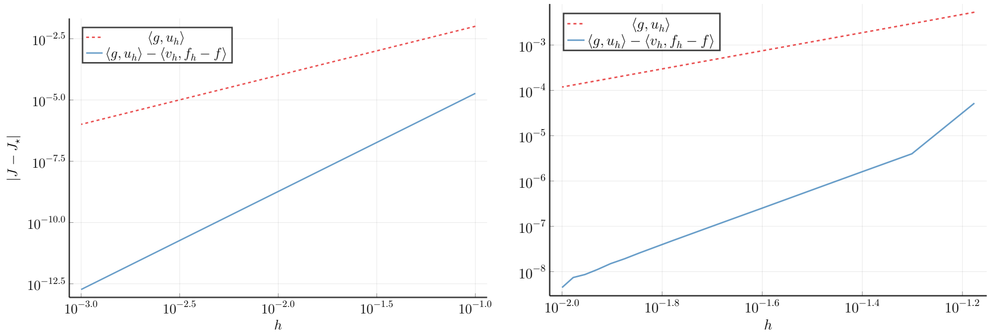

FIG. 7. *Functional evaluation errors for* (3.14)–(3.16) *(left) and* (3.18)–(3.19) *(right).*

that we are interested in the linear functional (3.15). The adjoint equation of (3.18), again obtained via integration by parts, reads

$$\kappa_1 \Delta v(x) - \kappa_2 \nabla \cdot v(x) = g(x) \qquad x \in \Omega, \tag{3.19a}$$

$$v(x) = 0 \qquad x \in \partial\Omega, \tag{3.19b}$$

and duality ensures (3.17).

In the case of heat transfer, $u(x)$ represents temperature and $f(x)$ sources or sinks. For example, with $g(x) = 1/\operatorname{vol}(\Omega)$, $J(u)$ represents the average temperature in Ω.

We choose $\Omega = [0, 1] \times [0, 1]$ and discretize (3.18) on a uniform $N \times N$ grid with the finite difference method such that the step along both coordinates is $h = 1/(N+1)$. With second-order central differences for first and second derivatives, the discretized

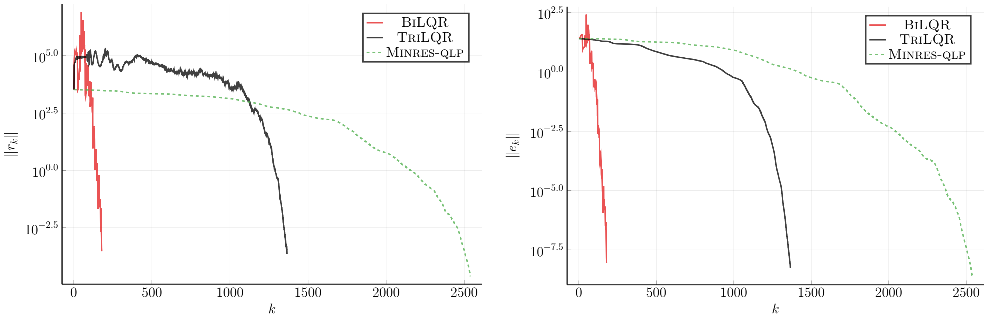

FIG. 8. *Residuals and errors norms of* BiLQR*,* TriLQR*, and* Minres-qlp *iterates for on* (3.18).

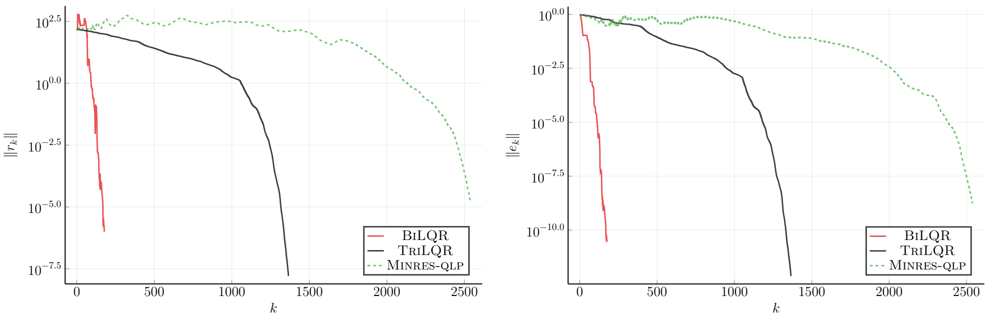

FIG. 9. *Residuals and errors norms of* BiLQR*,* TriLQR*, and* Minres-qlp *iterates on* (3.19).

operator has the structure

$$A = \begin{bmatrix} T & D_U & & \\ D_L & T & \ddots & \\ & \ddots & \ddots & D_U \\ & & D_L & T \end{bmatrix}, \quad T = \begin{bmatrix} -4\kappa_1 & \kappa_1 + \frac{1}{2}\kappa_2 h & & \\ \kappa_1 - \frac{1}{2}\kappa_2 h & -4\kappa_1 & \ddots & \\ & \ddots & \ddots & \kappa_1 + \frac{1}{2}\kappa_2 h \\ & & \kappa_1 - \frac{1}{2}\kappa_2 h & -4\kappa_1 \end{bmatrix},$$

$D_U = \operatorname{diag}(\kappa_1 + \frac{1}{2}\kappa_2 h)$, $D_L = \operatorname{diag}(\kappa_1 - \frac{1}{2}\kappa_2 h)$, where the right-hand sides b and c include the h^2 term. Solutions u_D and v_D contain an approximation of u and v at grid points stored column by column. The discretization of (3.19) with the same scheme yields A^T. We compare BiLQR, TriLQR, and Minres-qlp on (3.18) and (3.19) with $\kappa_1 = 5$, $\kappa_2 = 20$, $N = 50$, $g(x, y) = e^{x+y}$ and $f(x, y)$ such that the exact solution of (3.18) is $u_\star(x, y) = \sin(\pi x)\sin(\pi y)$. The resulting linear system has dimension 2,500 with 12,300 nonzeros. We use an absolute tolerance $\varepsilon_a = 10^{-10}$ and a relative tolerance $\varepsilon_r = 10^{-7}$ and terminate when both $\|r_k\| \le \varepsilon_a + \|b\|\varepsilon_r$ for (1.1) and $\|r_k\| \le \varepsilon_a + \|c\|\varepsilon_r$ for (3.2) hold.

Figures 8 and 9 report the evolution of the residual and error on (1.1) and (3.2) for (3.18) and (3.19), respectively. In this numerical illustration, residuals and errors are computed explicitly at each iteration as $b - Ax$, $c - A^T t$, $x - x_\star$, and $t - t_\star$ in order to discount errors in the approximation formulae for those expressions. In this example, BiLQR terminates in about six times fewer iterations than TriLQR and ten times fewer iterations than Minres-qlp. Only the Usymlq error and the Usymqr residual are monotonic. Although the Minres-qlp residual on (3.3) is monotonic, individual residuals on (1.1) and (3.2) are not.

We use bicubic spline interpolation and 3×3 points Gauss quadrature to compute estimates of $J(u)$ with and without correction term. With the $u_\star$ given above, $J_\star := J(u_\star) = (\pi(e+1))^2/(\pi^2+1)^2$. The right plot of Figure 7 illustrates the error in the evaluation of $J(u)$ as a function of h using the naive $J(u) \approx J(u_h)$ and improved (3.1) approximations.

4. Discussion. BiLQ completes the family of Krylov methods based on the Lanczos biorthogonalization process and is a natural companion to BiCG and Qmr. In general, neither the error nor the residual in the Euclidean norm is monotonic. However, BiLQ is a quasi-minimum error method, and its error is monotonic in the $U_p U_p^T$-norm, which is not iteration dependent but is unknown until the end of the biorthogonalization process. The same property holds for the Qmr residual.

Contrary to the Arnoldi (1951) and the Golub and Kahan (1965) processes, the Lanczos biorthogonalization and orthogonal tridiagonalization processes require two initial vectors. This distinguishing feature makes them readily suited to the simultaneous solution of primal and adjoint systems. A prime application is the superconvergent estimation of integral functionals in the context of discretized ODEs and PDEs. In our experiments, we observed that BiLQR outperforms both TriLQR and Minres-qlp applied to an augmented system in both error and residual norms.

Minres applied to (3.3) does not produce monotonic residuals in the individual primal and adjoint systems, as is apparent from the left plot of Figure 9. Herzog and Soodhalter (2017) devised a variant of Minres that allows block residuals to be monitored and that could be of interest in the context of estimating integral functionals.

Our Julia implementations of BiLQ, Qmr, Usymlq, Usymqr, BiLQR, TriLQR, and Minres-qlp is available at https://github.com/JuliaSmoothOptimizers/Krylov.jl and can be applied in any floating-point arithmetic supported by the language.

Acknowledgments. We sincerely thank Michael A. Saunders and two anonymous referees for numerous comments that improved the content and presentation of the present research.

REFERENCES

W. E. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Q. Appl. Math., 9 (1951), pp. 17–29, https://doi.org/10.1090/qam/42792.

J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, *Julia: A fresh approach to numerical computing*, SIAM Rev., 59 (2017), pp. 65–98, https://doi.org/10.1137/141000671.

A. Buttari, D. Orban, D. Ruiz, and D. Titley-Peloquin, *USYMLQR: A tridiagonalization method for symmetric saddle-point systems*, SIAM J. Sci. Comput., 41 (2019), pp. 409–432, https://doi.org/10.1137/18M1194900.

T. T. Chisholm and D. W. Zingg, *A Jacobian-free Newton-Krylov algorithm for compressible turbulent fluid flows*, J. Comput. Phys., 228 (2009), pp. 3490–3507, https://doi.org/10.1016/j.jcp.2009.02.004.

S.-C. 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 (2011), pp. 1810–1836, https://doi.org/10.1137/100787921.

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

T. A. Davis and E. P. Natarajan, *Sparse matrix methods for circuit simulation problems*, in Scientific Computing in Electrical Engineering SCEE 2010, Springer, New York, 2012, pp. 3–14, https://doi.org/10.1007/978-3-642-22453-9_1.

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

R. Estrin, D. Orban, and M. A. Saunders, *LNLQ: An iterative method for least-norm problems with an error minimization property*, SIAM J. Matrix Anal. Appl., 40 (2019b), pp. 1102–1124, https://doi.org/10.1137/18M1194948.

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

R. Fletcher, *Conjugate gradient methods for indefinite systems*, in Numerical Analysis, Springer, New York, 1976, pp. 73–89, https://doi.org/10.1007/BFb0080116.

R. W. Freund and N. M. Nachtigal, *QMR: A quasi-minimal residual method for non-Hermitian linear systems*, Numer. Math., 60 (1991), pp. 315–339, https://doi.org/10.1007/BF01385726.

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

G. H. Golub, M. Stoll, and A. Wathen, *Approximation of the scattering amplitude and linear systems*, ETNA, 31 (2008), pp. 178–203.

R. Herzog and K. Soodhalter, *A modified implementation of MINRES to monitor residual subvector norms for block systems*, SIAM J. Sci. Comput., 39 (2017), pp. A2645–A2663, https://doi.org/10.1137/16M1093021.

M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Natl. Bur. Stand., 49 (1952), pp. 409–436, https://doi.org/10.6028/jres.049.044.

M. Lai, *A note on finite difference discretizations for Poisson equation on a disk*, Numer. Methods Partial Differential Equations, 17 (2001), pp. 199–203, https://doi.org/10.1002/num.1.

C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Natl. Bur. Stand., 45 (1950), pp. 225–280, https://doi.org/10.6028/jres.045.026.

J. Lu and D. Darmofal, *A quasi-minimal residual method for simultaneous primal-dual solutions and superconvergent functional estimates*, SIAM J. Sci. Comput., 24 (2003), pp. 1693–1709, https://doi.org/10.1137/S1064827501390625.

C. C. Paige, I. Panayotov, and J.-P. M. Zemke, *An augmented analysis of the perturbed two-sided Lanczos tridiagonalization process*, Linear Algebra Appl., 447 (2014), pp. 119–132, https://doi.org/10.1016/j.laa.2013.05.009.

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

B. N. Parlett, D. R. Taylor, and Z. A. Liu, *A look-ahead Lanczos algorithm for unsymmetric matrices*, Math. Comp., 44 (1985), pp. 105–124, https://doi.org/10.2307/2007796.

N. A. Pierce and M. B. Giles, *Adjoint recovery of superconvergent functionals from PDE approximations*, SIAM Rev., 42 (2000), pp. 247–264, https://doi.org/10.1137/S0036144598349423.

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

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

P. Sonneveld, *CGS, a fast Lanczos-type solver for nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 10 (1989), pp. 36–52, https://doi.org/10.1137/0910004.

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. Stat. Comput., 13 (1992), pp. 631–644, https://doi.org/10.1137/0913035.

R. Weiss, *Error-minimizing Krylov subspace methods*, SIAM J. Sci. Comput., 15 (1994), pp. 511–527, https://doi.org/10.1137/0915034.