

## A KRYLOV SUBSPACE METHOD FOR QUADRATIC MATRIX POLYNOMIALS WITH APPLICATION TO CONSTRAINED LEAST SQUARES PROBLEMS\*

REN-CANG LI<sup>†</sup> AND QIANG YE<sup>†</sup>

**Abstract.** We present a Krylov subspace–type projection method for a quadratic matrix polynomial  $\lambda^2 I - \lambda A - B$  that works directly with  $A$  and  $B$  without going through any linearization. We discuss a special case when one matrix is a low rank perturbation of the other matrix. We also apply the method to solve quadratically constrained linear least squares problem through a reformulation of Gander, Golub, and von Matt as a quadratic eigenvalue problem, and we demonstrate the effectiveness of this approach. Numerical examples are given to illustrate the efficiency of the algorithms.

**Key words.** quadratic matrix polynomial, Krylov subspace, quadratic eigenvalue problem, least squares problem, quadratic constraint

**AMS subject classifications.** 65F15, 65F20, 15A18

**DOI.** 10.1137/S0895479802409390

**1. Introduction.** Krylov subspace techniques are widely used for solving linear systems of equations and eigenvalue problems involving large and sparse matrices [7, 14]. It has found applications in many other large scale matrix problems such as model reductions of linear input-output systems. The basic idea of the techniques is to extract information of an  $n \times n$  matrix  $A$  most relevant to the underlying computational problem through utilizing the so-called *Krylov* subspace

$$\mathcal{K}_k(A, v) = \text{span}\{v, Av, \dots, A^{k-1}v\}$$

or through utilizing two (row and column) Krylov subspaces  $\mathcal{K}_k(A, v)$  and  $\mathcal{K}_k(A^*, w)$  simultaneously, where  $v$  and  $w$  are vectors of dimension  $n$  and  $A^*$  is the conjugate transpose. This is realized by the *Lanczos/Arnoldi process* [1, 18]. See also [7, 14, 22, 28, 29].

The quadratic eigenvalue problem (QEP) in its generality takes the form

$$(1.1) \quad (\lambda^2 M + \lambda C + K)z = 0,$$

where  $M, C, K$  are  $n \times n$  matrices, scalar  $\lambda$  is called an *eigenvalue*, and  $n$ -dimensional  $0 \neq z$  is a corresponding (right) *eigenvector*. In solving it when  $n$  is large and  $M, C, K$  are sparse, it is often transformed implicitly into a mathematically equivalent *monic* QEP

$$(1.2) \quad (\lambda^2 I_n - \lambda A - B)x = 0,$$

where  $A$  and  $B$  stay in some factored forms so that the matrix-vector multiplications by  $A$  and  $B$  are cheap. (It is possible that  $\lambda$  in (1.2) differs from the one in the

---

\*Received by the editors June 7, 2002; accepted for publication (in revised form) by H. A. van der Vorst March 10, 2003; published electronically August 19, 2003.

<http://www.siam.org/journals/simax/25-2/40939.html>

<sup>†</sup>Department of Mathematics, University of Kentucky, Lexington, KY 40506 (rcli@ms.uky.edu, qye@ms.uky.edu). The research of the first author was supported in part by the National Science Foundation CAREER award under grant CCR-9875201 and by the National Science Foundation under grant ACI-9721388. The research of the second author was supported in part by the National Science Foundation under grant CCR-0098133.

original (1.1) but relates to it by a shifting transformation.) For this reason, we shall focus in this paper on monic QEPs.

A related problem is the approximation of the transfer function

$$f(s) = c^*(s^2 I_n - sA - B)^{-1}b,$$

which arises in a single input single output system as governed by a second order initial value problem.

For these problems, a typical approach is to reduce them to an equivalent linear problem for the  $2n \times 2n$  matrix [13],

$$A_{\text{LIN}} = \begin{pmatrix} 0 & I \\ B & A \end{pmatrix},$$

to which well-established methods can be applied (e.g., ARPACK [19]). This is called *linearization*. For the eigenvalue problem or the model reduction problem, one can use the Lanczos or the Arnoldi algorithm to produce a small projection of  $A_{\text{LIN}}$  on a Krylov subspace, which is then used to approximate  $A_{\text{LIN}}$ . This, however, increases the computational complexity by doubling the problem size. Furthermore, the projection of  $A_{\text{LIN}}$  is usually not a linearization of any QEP and thus loses its intrinsic physical connection to the problem that it approximates. As a result, for example, certain spectral properties of the original problem are not preserved in the projection and the approximations so obtained may not possess certain desirable properties such as the Galerkin condition. For the model reduction problem, the reduced model that is obtained by applying the Arnoldi or the Lanczos process to the linearization problem  $A_{\text{LIN}}$  cannot be synthesized with a physical model of QEP [2].

It is thus desirable to approximate a large scale QEP with another QEP of smaller size. The objective of this paper is to extend the *standard Arnoldi process* (and the *standard Lanczos process*) to cover matrix polynomials without going through any linearization. Namely, we develop a Krylov-type projection process applied simultaneously to  $A$  and  $B$  so as to obtain a projected lower-dimensional matrix polynomial to approximate the original one. With two matrices involved, the projections will no longer be in the upper Hessenberg (or tridiagonal) form, but rather a lower banded form with a growing lower bandwidth as the process progresses. However, in the case when some combination of the coefficient matrices  $A$  and  $B$  is of low rank, the projection matrix simplifies to a banded form and the algorithm becomes more efficient. We note that several other methods [20, 25] have been developed that do not rely on the linearization processes (see also [3, 30]).

As an application, we shall study the following quadratically constrained least squares problem

$$(1.3) \quad \min_{\|x\|_2=\delta} \|Cx - b\|_2,$$

which arises, for example, in the regularization solution of discretized ill-posed problem (see [15, 16] and [23]), where all numbers are real,  $C$  is  $m \times n$ , and  $x$  and  $b$  are vectors of dimensions  $n$  and  $m$ , respectively. It can be formulated as the constrained minimization problem

$$\min_{x^T x = \delta^2} x^T H x - 2g^T x,$$

where  $H = C^T C$ ,  $g = C^T b$ , and  $C^T$  is the transpose of  $C$ . A slightly more general form that uses the inequality constraint  $x^T x \leq \delta^2$  is called a trust region subproblem

(see [23], for example). We note that the problem with the inequality constraint will be more general (i.e., it will have no solution satisfying the equality constraint) only when  $H$  is invertible and  $x = H^{-1}g$  (the solution to the unconstrained problem) lies in the interior of the constraint region [21]. To solve the above constrained minimization problem, several factorization-based methods have been developed [9, 11, 12, 21, 26], which typically apply to small or moderate size problems. For large problems, however, iterative methods are usually considered; see [4, 5, 6, 16, 23, 24, 27] for various methods developed.

In [11], Gander, Golub, and von Matt show that the above minimization problem can be transformed to the QEP

$$(\lambda^2 I - 2\lambda H + H^2 - \delta^{-2} g g^T) y = 0.$$

With the structure of this eigenvalue problem, the Krylov-type method can be adapted to solve it efficiently. This turns out to be a very efficient approach for solving the above constrained minimization problem, and the process of the Krylov-type method itself has a regularization effect for discrete ill-posed problems (1.3). We shall discuss various theoretical and numerical issues concerning this approach.

The paper is organized as follows. We present the Arnoldi-type algorithm for the quadratic matrix polynomial in section 2 and then the low rank perturbed case in section 3. We study the constrained least squares problem via the Arnoldi-type algorithm in section 4. We present some numerical examples in section 5 to illustrate the efficiency of the algorithms, and we give our concluding remarks in section 6.

**Notation.** Throughout,  $\|\cdot\|$  refers to the 2-norm, i.e.,  $\|v\|^2 = v^*v$ .  $I_n$  is the  $n \times n$  identity matrix or simply  $I$  whenever its dimension is clear from the context;  $e_j$  is its  $j$ th column.  $\lambda(X)$  is the spectrum of  $X$ . We use MATLAB-like notation  $X_{(i:j,k:\ell)}$  to denote the submatrix of  $X$ , consisting of the intersections of rows  $i$  to  $j$  and columns  $k$  to  $\ell$ , and when  $i : j$  is replaced by  $:$ , it means all rows, similarly for columns. We shall use generic notation  $x$  for a possibly nonzero scalar or vector and  $X$  for a possibly nonzero matrix.

**2. Arnoldi-type process for monic quadratic matrix polynomials.** We first develop an Arnoldi-type process for monic quadratic matrix polynomial  $I\lambda^2 - A\lambda - B$ . Our algorithm will be based on a simultaneous orthogonal reduction of  $A$  and  $B$ . For the sake of generality, we state all results in the field of complex numbers. However, when all numbers involved are real, the only changes needed to be made are to replace  $\mathbb{C}$  by  $\mathbb{R}$  and asterisk superscripts  $*$  by  $.^T$ .

**2.1. Decomposition theorem.** Our proofs below rely on the ability to transform a vector to a scalar multiplier of  $e_1$  by an orthogonal transformation. This can be realized by at least two ways: by a Householder transformation or by a sequence of Givens rotations [7, 14, 31].

LEMMA 2.1. *There is a unitary matrix  $Q \in \mathbb{C}^{n \times n}$  with  $Qe_1 = e_1$  such that*

$$Q^* A Q = H_a \equiv (h_{a;ij}), \quad Q^* B Q = H_b \equiv (h_{b;ij})^1$$

*satisfy*

$$h_{a;ij} = 0 \text{ for } i \geq 2j + 1, \quad h_{b;ij} = 0 \text{ for } i \geq 2j + 2.$$

---

<sup>1</sup> $h_{a;ij}$  denotes the  $(i, j)$  entry of  $H_a$ , but we shall also use  $h_{a;i,j}$  to denote the same when  $i$  and  $j$  are not clearly separated.

*Proof.* Our proof is constructive. It goes as follows. Partition

$$A = \begin{matrix} & 1 & n-1 \\ \begin{matrix} 1 \\ n-1 \end{matrix} & \left( \begin{array}{c|c} a_{11} & \mathbf{x} \\ a_1 & \mathbf{X} \end{array} \right), \end{matrix}$$

and then find a unitary  $\widehat{Q}_{1a} \in \mathbb{C}^{(n-1) \times (n-1)}$  such that  $\widehat{Q}_{1a}^* a_1 = \alpha_1 e_1$ . Let  $Q_{1a} = \text{diag}(1, \widehat{Q}_{1a})$ . We have

$$Q_{1a}^* A Q_{1a} = \left( \begin{array}{c|c} a_{11} & \mathbf{x} \\ \alpha_1 & \mathbf{X} \\ 0 & \end{array} \right), \quad Q_{1a}^* B Q_{1a} = \begin{matrix} & 1 & n-1 \\ \begin{matrix} 1 \\ n-2 \end{matrix} & \left( \begin{array}{c|c} b_{11} & \mathbf{x} \\ b_{21} & \mathbf{x} \\ b_1 & \mathbf{X} \end{array} \right). \end{matrix}$$

Now find a unitary  $\widehat{Q}_{1b} \in \mathbb{C}^{(n-2) \times (n-2)}$  such that  $\widehat{Q}_{1b}^* b_1 = \beta_1 e_1$ . Let  $Q_{1b} = \text{diag}(I_2, \widehat{Q}_{1b})$  and  $Q_1 \stackrel{\text{def}}{=} Q_{1a} Q_{1b}$ . We have

$$Q_1^* A Q_1 = \left( \begin{array}{c|c} a_{11} & \mathbf{x} \\ \alpha_1 & \mathbf{X} \\ 0 & \end{array} \right), \quad Q_1^* B Q_1 = \left( \begin{array}{c|c} b_{11} & \mathbf{x} \\ b_{21} & \mathbf{x} \\ \beta_1 & \mathbf{X} \\ 0 & \end{array} \right).$$

This puts the first columns of  $A$  and  $B$  into the desired forms. Next we work on their second columns. Partition

$$Q_1^* A Q_1 = \begin{matrix} & 1 & 1 & n-2 \\ \begin{matrix} 1 \\ 1 \\ n-3 \end{matrix} & \left( \begin{array}{c|c|c} \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & a_{32} & \mathbf{x} \\ 0 & a_2 & \mathbf{X} \end{array} \right), \end{matrix}$$

and then find a unitary  $\widehat{Q}_{2a} \in \mathbb{C}^{(n-3) \times (n-3)}$  such that  $\widehat{Q}_{2a}^* a_2 = \alpha_2 e_1$ . Let  $Q_{2a} = \text{diag}(I_3, \widehat{Q}_{2a})$ . We have

$$Q_{2a}^* Q_1^* A Q_1 Q_{2a} = \left( \begin{array}{c|c|c} \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & a_{32} & \mathbf{x} \\ 0 & \alpha_2 & \mathbf{X} \\ & 0 & \end{array} \right), \quad Q_{2a}^* Q_1^* B Q_1 Q_{2a} = \begin{matrix} & 1 & 1 & n-2 \\ \begin{matrix} 1 \\ 1 \\ 1 \\ n-4 \end{matrix} & \left( \begin{array}{c|c|c} \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & b_{32} & \mathbf{x} \\ 0 & b_{42} & \mathbf{x} \\ 0 & b_2 & \mathbf{X} \end{array} \right). \end{matrix}$$

Now find a unitary  $\widehat{Q}_{2b} \in \mathbb{C}^{(n-4) \times (n-4)}$  such that  $\widehat{Q}_{2b}^* b_2 = \beta_2 e_1$ . Let  $Q_{2b} = \text{diag}(I_4, \widehat{Q}_{2b})$  and  $Q_2 \stackrel{\text{def}}{=} Q_{2a} Q_{2b}$ . We have

$$Q_2^* Q_1^* A Q_1 Q_2 = \left( \begin{array}{c|c|c} \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & a_{32} & \mathbf{x} \\ 0 & \alpha_1 & \mathbf{X} \\ & 0 & \end{array} \right), \quad Q_2^* Q_1^* B Q_1 Q_2 = \left( \begin{array}{c|c|c} \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & b_{32} & \mathbf{x} \\ 0 & b_{42} & \mathbf{x} \\ 0 & \beta_2 & \mathbf{X} \\ & 0 & \end{array} \right).$$

By now the first two columns of  $A$  and  $B$  are put into the desired forms. The process proceeds in a similar fashion from here. At the end, the  $j$ th column of transformed  $A$  has  $2j$  possible nonzero entries at the top, and the  $j$ th column of transformed  $B$  has  $2j + 1$  possible nonzero entries also at the top. Taking  $Q = Q_1 Q_2 \cdots Q_k$  completes the reduction, where at most  $k \leq n/2$ . It is easy to see  $Qe_1 = e_1$ .  $\square$

**THEOREM 2.2.** *Given  $q_1 \in \mathbb{C}^n$  with  $\|q_1\|_2 = 1$ , there is a unitary matrix  $Q \in \mathbb{C}^{n \times n}$  with  $Qe_1 = q_1$  such that*

$$(2.1) \quad Q^*AQ = H_a \equiv (h_{a;ij}), \quad Q^*BQ = H_b \equiv (h_{b;ij})$$

satisfy

$$(2.2) \quad h_{a;ij} = 0 \text{ for } i \geq 2j + 1, \quad h_{b;ij} = 0 \text{ for } i \geq 2j + 2.$$

*Proof.* Find a unitary  $Q_0 \in \mathbb{C}^{n \times n}$  with  $Q_0e_1 = q_1$ . Then apply Lemma 2.2 to  $Q_0^*AQ_0$  and  $Q_0^*BQ_0$  to get a unitary  $\widehat{Q} \in \mathbb{C}^{n \times n}$  with  $\widehat{Q}e_1 = e_1$  such that

$$\widehat{Q}^*(Q_0^*AQ_0)\widehat{Q} \equiv H_a, \quad \widehat{Q}^*(Q_0^*BQ_0)\widehat{Q} \equiv H_b$$

have the desired forms. Now letting  $Q = Q_0\widehat{Q}$  completes the proof.  $\square$

**2.2. Arnoldi-type process.** Although the proofs for Lemma 2.1 and Theorem 2.2 are constructive, they are of little use when it comes to numerical computations with large and sparse  $A$  and  $B$  for which we can only afford to generate  $Q$ ,  $H_a$ , and  $H_b$  partially. In what follows, we shall present an Arnoldi-type process to do so. Rewrite (2.1) to get

$$(2.3) \quad AQ = QH_a, \quad BQ = QH_b.$$

Inspecting the  $j$ th column, we see

$$(2.4) \quad Aq_j = \sum_{i=1}^{2j-1} q_i h_{a;ij} + q_{2j} h_{a;2j,j},$$

$$(2.5) \quad Bq_j = \sum_{i=1}^{2j} q_i h_{b;ij} + q_{2j+1} h_{b;2j+1,j}.$$

Equation (2.4) and the orthogonality among  $q_1, \dots, q_{2j}$  yield

$$h_{a;ij} = q_i^* Aq_j \quad \text{for } i \leq 2j - 1,$$

and then we have

$$h_{a;2j,j} = \left\| Aq_j - \sum_{i=1}^{2j-1} q_i h_{a;ij} \right\|_2,$$

$$q_{2j} = \left( Aq_j - \sum_{i=1}^{2j-1} q_i h_{a;ij} \right) / h_{a;2j,j},$$

where we assume also  $h_{a;2j,j} \neq 0$ . Similarly, (2.5) implies

$$h_{b;ij} = q_i^* Bq_j \quad \text{for } i \leq 2j,$$

and then

$$h_{b;2j+1,j} = \left\| Bq_j - \sum_{i=1}^{2j} q_i h_{b;i,j} \right\|_2,$$

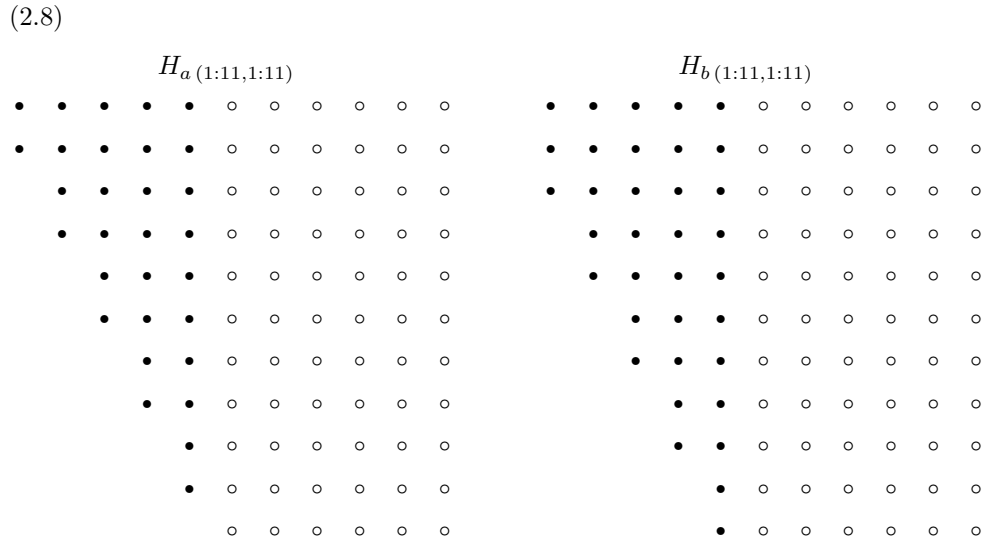
$$q_{2j+1} = \left( Bq_j - \sum_{i=1}^{2j} q_i h_{b;i,j} \right) / h_{b;2j+1,j},$$

where we assume  $h_{b;2j+1,j} \neq 0$ . This leads to a process that constructs  $q_{2j}, q_{2j+1}$  from  $q_1, q_2, \dots, q_{2j-1}$ . After  $k$  steps of construction, we obtain  $q_1, q_2, \dots, q_{2k+1}$  such that

(2.6)  $AQ_{(:,1:k)} = Q_{(:,1:2k)}H_a(1:2k,1:k),$

(2.7)  $BQ_{(:,1:k)} = Q_{(:,1:2k+1)}H_b(1:2k+1,1:k).$

The following figures in (2.8) show what the computed parts of  $H_a$  and  $H_b$  look like for  $k = 5$ , where the entries marked by *unfilled circles* are not computed yet.



With those computed entries,  $H_a(1:k,1:k)$  and  $H_b(1:k,1:k)$  provide the projections of  $A$  and  $B$  on  $\text{span}\{Q_{(:,1:k)}\}$ . To fully utilize those unused computed entries, we can complete  $H_a(1:2k+1,1:2k+1)$  and  $H_b(1:2k+1,1:2k+1)$  by computing

$$h_{a;ij} = q_i^* Aq_j, \quad h_{b;ij} = q_i^* Bq_j$$

for  $1 \leq i \leq 2k + 1$  and  $k + 1 \leq j \leq 2k + 1$  (i.e., the entries marked by *unfilled circles* above), which will then give the projections on a bigger subspace  $\text{span}\{Q_{(:,1:2k+1)}\}$ . This requires computing  $Aq_j$  and  $Bq_j$  for  $k + 1 \leq j \leq 2j + 1$ . Therefore, to construct a  $(2k + 1) \times (2k + 1)$  projection, we still need  $2k + 1$  matrix-vector multiplications by both  $A$  and  $B$ , but the number of vector operations required will be less.

So far, we have assumed that  $h_{a;2j,j}$  and  $h_{b;2j+1,j}$  are nonzero. When an  $h_{a;2j,j}$  or  $h_{b;2j+1,j}$  vanishes, no new  $q$ -vector can be generated, but we will show that the process can be continued. This is actually a welcome situation.

In the process, we apply  $A$  and  $B$  alternately on each vector in the sequence to construct new  $q$ -vectors. At any given point, let  $N$  be the number of  $q$ -vectors already

constructed. At the beginning of the process,  $N = 1$  and there is only  $q_1$ , which has not yet been applied by  $A$  and  $B$ . For the first step ( $j = 1$ ), we apply  $A$  to  $q_1$ , which may or may not generate a new  $q$ -vector, and if it does,  $N \leftarrow N + 1$  (which is 2) and  $q_N$  is constructed. We then apply  $B$  to  $q_1$ , which again may or may not generate a new  $q$ -vector, and if it does,  $N \leftarrow N + 1$  (which is either 2 or 3) and we have constructed a new  $q_N$ . Then,  $N$   $q$ -vectors have been constructed, and if  $N = 1$ , the process can be terminated with  $\text{span}\{q_1\}$  being invariant under both  $A$  and  $B$ . If  $N \geq 2$ , we then proceed to apply  $A$  and  $B$  to  $q_2$  in the same way. In general, at the beginning of step  $j$ , among  $q_1, \dots, q_N$  that have been constructed,  $q_1, \dots, q_{j-1}$  have been applied by  $A$  and  $B$ . If  $N = j - 1$ ,  $\text{span}\{q_1, \dots, q_N\}$  is invariant under both  $A$  and  $B$  and we can terminate the process. If  $N \geq j$ , we apply  $A$  to  $q_j$  (the next vector that has not been applied yet), and if a new vector is generated,  $N \leftarrow N + 1$  and  $q_N$  is added to the  $q$ -vector list. We then apply  $B$  to  $q_j$  similarly. The process continues until  $N = j - 1$ , which must occur at  $j = n + 1$ , or a preselected  $k$  number of steps is reached. Thus,  $N$  may be much smaller than  $2k + 1$ . To fully utilize the information provided by the generated subspace  $\text{span}\{Q(:, 1 : N)\}$ , in our later numerical examples we compute the fully projected  $H_{a(1:N, 1:N)}$  and  $H_{b(1:N, 1:N)}$ . Algorithm 2.1 summarizes our new process.

ALGORITHM 2.1 (Arnoldi-type process).

1. Given  $q_1$  with  $\|q_1\|_2 = 1$ ;
2.  $N = 1$ ;
3. For  $j = 1, 2, \dots, k$  do
4.     If  $j > N$ , **BREAK**;
5.      $\hat{q} = Aq_j$ ;
6.     For  $i = 1, 2, \dots, N$  do
7.          $h_{a;ij} = q_i^* \hat{q}$ ;  $\hat{q} = \hat{q} - q_i h_{a;ij}$ ;
8.     EndDo
9.      $h_{a;N+1,j} = \|\hat{q}\|_2$ ;
10.     If  $h_{a;N+1,j} > 0$ ,
11.          $N = N + 1$ ,  $q_N = \hat{q}/h_{a;Nj}$ ;
12.     EndIf
13.      $\hat{q} = Bq_j$ ;
14.     For  $i = 1, 2, \dots, N$  do
15.          $h_{b;ij} = q_i^* \hat{q}$ ;  $\hat{q} = \hat{q} - q_i h_{b;ij}$ ;
16.     EndDo
17.      $h_{b;N+1,j} = \|\hat{q}\|_2$ ;
18.     If  $h_{b;N+1,j} > 0$ ,
19.          $N = N + 1$ ,  $q_N = \hat{q}/h_{a;N,j}$ ;
20.     EndIf
21. EndDo

We point out that an appropriate tolerance must be used in practical implementations of line 10 and line 18 of Algorithm 2.1 as, e.g.,  $h_{a;N+1,j} > n\epsilon\|A\|$  and  $h_{b;N+1,j} > n\epsilon\|B\|$ , where  $\epsilon$  is the machine roundoff unit. Define

$$(2.9) \quad \alpha_j = \text{value of } N \text{ at line 12 at step } j,$$

$$(2.10) \quad \beta_j = \text{value of } N \text{ at line 20 at step } j,$$

with  $\alpha_0 = \beta_0 = 1$ . Then,

$$Aq_j = \sum_{i=1}^{\alpha_j} h_{a;ij} q_i, \quad Bq_j = \sum_{i=1}^{\beta_j} h_{b;ij} q_i.$$

Thus, upon completion of the above process, we have in general

$$(2.11) \quad A Q_{(:,1:k)} = Q_{(:,1:\alpha_k)} H_a(1:\alpha_k, 1:k),$$

$$(2.12) \quad B Q_{(:,1:k)} = Q_{(:,1:\beta_k)} H_b(1:\beta_k, 1:k),$$

unless the  $j$ -loop is forced to **BREAK** out at line 4, in which case we have obtained an invariant subspace of both  $A$  and  $B$  with

$$(2.13) \quad A Q_{(:,1:N)} = Q_{(:,1:N)} H_a(1:N, 1:N),$$

$$(2.14) \quad B Q_{(:,1:N)} = Q_{(:,1:N)} H_b(1:N, 1:N),$$

where  $N$  takes its value when the  $j$ -loop is terminated.

It is clear that

$$\beta_{j-1} \leq \alpha_j \leq \beta_{j-1} + 1 \text{ and } \alpha_j \leq \beta_j \leq \alpha_j + 1.$$

Furthermore, the nonzeros of the  $j$ th column of  $H_a$  (and  $H_b$ , resp.) are contained in the first  $\alpha_j$  ( $\beta_j$ , resp.) entries only.  $\alpha_j$  (and  $\beta_j$  as well) can increase at most by 2 at each step. So, the nonzero patterns in  $H_a$  and  $H_b$  are contained in those as described in (2.8).

We can use the reduced matrices  $H_a(1:N, 1:N)$  and  $H_b(1:N, 1:N)$  to approximate  $A$  and  $B$ . For example, we can use the eigenvalues of  $\lambda^2 I - \lambda H_a(1:N, 1:N) - H_b(1:N, 1:N)$  to approximate those of the original quadratic problem. However, as the lower bandwidth of  $H_a$  and  $H_b$  grows very fast in general, the convergence is expected to be slow in general; see [17] for an analysis on the relation between the bandwidth and the speed of convergence. There are some special cases where the lower bandwidth can be bounded by a constant or grows at a much slower pace than in general. We shall discuss two such cases in the next two sections.

Similar to our derivation here, a (nonsymmetric) Lanczos-type process can be derived. The details will be presented in [17]. Finally, we remark that the way that the subspace  $\text{span}\{q_1, \dots, q_N\}$  are generated here bears some resemblance to the so-called generalized Krylov subspace in [33].

**2.3. Hermitian case.** When  $A$  and  $B$  are Hermitian,  $H_a$  and  $H_b$  will also be Hermitian. In that case, their upper triangular parts need not be computed and it is easy to prove that the recurrences are simplified to

$$h_{a;\alpha_j j} q_{\alpha_j} = A q_j - \sum_{1 \leq i < \alpha_j, \text{ and } \alpha_i \geq j} h_{a;i j} q_i,$$

$$h_{b;\beta_j j} q_{\beta_j} = B q_j - \sum_{1 \leq i < \beta_j, \text{ and } \beta_i \geq j} h_{b;i j} q_i.$$

We call the corresponding algorithm the symmetric Lanczos-type process. We omit the details here.

It is worth mentioning that the reduction process here also preserves other structural properties such as skew-symmetry or positive-definiteness in  $A$  or  $B$ .

**3. Low rank case.** In this section, we consider the case when some linear combination of  $A$  and  $B$  is of low rank, i.e.,

$$\zeta B + \xi A = E,$$

where  $E$  is a matrix of rank  $p$  and  $\zeta$  and  $\xi$  are some, possibly unknown, scalars, at least one of which is nonzero. This includes the cases when one matrix is of low rank or is a low rank perturbation of the other matrix. We show that the Arnoldi-type process will be greatly simplified to yield a reduction with a lower bandwidth at most  $p + 1$  throughout the process. The resulting algorithm will be much more efficient.

Apply the Arnoldi-type process (Algorithm 2.1), we obtain at step  $k$  (see (2.11) and (2.12))

$$\begin{aligned} AQ_{(:,1:k)} &= Q_{(:,1:\alpha_k)} H_{a(1:\alpha_k,1:k)} = Q_{(:,1:\beta_k)} H_{a(1:\beta_k,1:k)}, \\ BQ_{(:,1:k)} &= Q_{(:,1:\beta_k)} H_{b(1:\beta_k,1:k)}. \end{aligned}$$

Therefore,

$$EQ_{(:,1:k)} = Q_{(:,1:\beta_k)} (\zeta H_{b(1:\beta_k,1:k)} + \xi H_{a(1:\beta_k,1:k)}).$$

This shows  $\zeta H_{b(1:\beta_k,1:k)} + \xi H_{a(1:\beta_k,1:k)}$  has at most rank  $p$ . We consider now the case that  $\zeta \neq 0$ ; the case that  $\xi \neq 0$  follows similarly. From the structures of  $H_a$  and  $H_b$ , it can be seen that there are at most  $p$  columns in which  $H_b$  has more nonzeros than  $H_a$ , which is the time in the process that the lower bandwidth is increased. Thus, the lower bandwidth of  $H_a$  and  $H_b$  can grow at most  $p$  times throughout the process and is therefore bounded by  $p + 1$ . To be more rigorous, let  $i_1 < i_2 < \dots < i_\ell$  be the index  $j$  between 1 and  $k$  such that  $\beta_j = \alpha_j + 1$ , in which case  $h_{b;\beta_j,j} \neq 0$ . For such  $j \in \{i_1, i_2, \dots, i_\ell\}$ ,  $\beta_j > \alpha_j \geq \beta_{j-1}$  and therefore  $h_{a;\beta_j,j} = 0$ . Furthermore,  $\beta_{i_1} < \beta_{i_2} < \dots < \beta_{i_\ell}$ . It follows from examining the  $i_1, i_2, \dots, i_\ell$ th columns of  $\zeta H_{b(1:\beta_k,1:k)} + \xi H_{a(1:\alpha_k,1:k)}$  that its rank is at least  $\ell$ . Thus,

$$\ell \leq \text{rank}(EQ_{(:,1:k)}) \leq p.$$

This demonstrates that there are at most  $p$  indexes  $j$  for which  $\beta_j = \alpha_j + 1$ . Hence there are at most  $p$  indexes  $j$  for which  $\alpha_{j+1} = \alpha_j + 2$ . For the same reason, there are at most  $p$  indexes  $j$  for which  $\beta_{j+1} = \beta_j + 2$ . Thus,

$$\alpha_j \leq j + 1 + p \text{ and } \beta_j \leq j + 1 + p.$$

So,  $H_{a(1:\alpha_k,1:k)}$  and  $H_{b(1:\beta_k,1:k)}$  are banded matrices with lower bandwidth at most  $p + 1$ . We state this result as the following theorem.

**THEOREM 3.1.** *In Algorithm 2.1, if  $\zeta B + \xi A = E$  (either  $\zeta$  or  $\xi \neq 0$ ) and  $E$  is a matrix of rank  $p$ , then  $\alpha_j \leq j + 1 + p$  and  $\beta_j \leq j + 1 + p$ . In particular,  $H_{a(1:\alpha_k,1:k)}$  and  $H_{b(1:\beta_k,1:k)}$  are banded with lower bandwidth at most  $p + 1$ .*

We note that it is not necessary to know the explicit combination  $\zeta B + \xi A = E$  or the rank of  $E$  in advance. The algorithm will produce a reduction with the lower bandwidth limited by the rank of  $E$ . In practice, we may need to implement some reorthogonalization technique and use an appropriate tolerance in line 10 and line 18 of Algorithm 2.1. Then, the lower bandwidth will also be limited by the rank of  $E$  (see numerical examples in subsection 5.1).

**3.1. Quadratic eigenvalue problems.** The Arnoldi-type method can be used to find some eigenvalues and eigenvectors of the quadratic matrix polynomial  $I\lambda^2 - A\lambda - B$ . If Algorithm 2.1 produces  $Q_{(:,1:k)}$ ,  $H_{a(1:k,1:k)}$ , and  $H_{b(1:k,1:k)}$ , let  $\theta$  be an eigenvalue and  $u$  a right eigenvector of

$$(3.1) \quad I\lambda^2 - H_{a(1:k,1:k)}\lambda - H_{b(1:k,1:k)}.$$

We use  $(\theta, y)$  as an approximate eigenvalue and eigenvector for the original problem, where

$$(3.2) \quad y = Q_{(:,1:k)}u.$$

$\theta$  will be called a Ritz value and  $y$  a Ritz vector. We note that the method works for general  $A$  and  $B$ , but the convergence may be slow [17]. For this reason, we shall consider the current case that  $\zeta B + \xi A = E$  is of low rank.

In the next theorem, we present an a posteriori residual bound and show that the Ritz values and the Ritz vectors satisfy a Galerkin-type condition.

**THEOREM 3.2.** *Let  $H_a(1:k,1:k)$  and  $H_b(1:k,1:k)$  be obtained from  $k$  steps of the Arnoldi-type process (Algorithm 2.1), and let  $\theta$  be an eigenvalue and  $u$  be a unit right eigenvector of (3.1). Then the Ritz value  $\theta$  and the Ritz vector  $y = Q_{(:,1:k)}u$  satisfy the following Galerkin-type condition:*

$$(3.3) \quad r \equiv (\theta^2 I - \theta A - B)y \perp \text{span}\{Q_{(:,1:k)}\}.$$

Furthermore,

$$\|r\| \leq (|\theta| \|A\| + \|B\|)\|u_{(k-p:k)}\|.$$

*Proof.* First, from (2.6) and (2.7), we have

$$\begin{aligned} AQ_{(:,1:k)} &= Q_{(:,1:k)}H_a(1:k,1:k) + Q_{(:,k+1:k+1+p)}H_a(k+1:k+1+p,1:k), \\ BQ_{(:,1:k)} &= Q_{(:,1:k)}H_b(1:k,1:k) + Q_{(:,k+1:k+1+p)}H_b(k+1:k+1+p,1:k). \end{aligned}$$

Then

$$\begin{aligned} r &= (\theta^2 Q_{(:,1:k)} - \theta AQ_{(:,1:k)} - BQ_{(:,1:k)})u \\ &= Q_{(:,1:k)}(\theta^2 I - \theta H_a(1:k,1:k) - H_b(1:k,1:k))u \\ &\quad - \theta Q_{(:,k+1:k+1+p)}H_a(k+1:k+1+p,1:k)u - Q_{(:,k+1:k+1+p)}H_b(k+1:k+1+p,1:k)u \\ &= -Q_{(:,k+1:k+1+p)}(\theta H_a(k+1:k+1+p,k-p:k) + H_b(k+1:k+1+p,k-p:k))u_{(k-p:k)}. \end{aligned}$$

The orthogonality among  $q$ -vectors implies (3.3). Furthermore,

$$\|H_a(k+1:k+1+p,k-p:k)\| = \|Q_{(k+1:k+1+p,:)}^* A Q_{(:,k-p:k)}\| \leq \|A\|.$$

Similarly,  $\|H_b(k+1:k+1+p,k-p:k)\| \leq \|B\|$ . Taking the norm on  $r$  above, we obtain the bound.  $\square$

The theorem shows that if the last  $p+1$  entries of an approximate eigenvector  $u$  become small, then the corresponding approximate eigenvalue will be a good approximation. This is usually the case for extreme eigenvalues of tridiagonal matrices produced by the standard Lanczos algorithm, and we observe that the banded matrices here appear to have a similar property.

We next derive an a priori convergence analysis similar to that of [32]. Here, we establish a relationship between the Ritz values and the eigenvalues of the original QEP through the linearizations. Let

$$L = \begin{pmatrix} 0 & I \\ H_b & H_a \end{pmatrix} \quad \text{and} \quad L_k = \begin{pmatrix} 0 & I \\ H_b(1:k,1:k) & H_a(1:k,1:k) \end{pmatrix},$$

where  $H_a$  and  $H_b$  are  $n \times n$  as obtained by continuing the reduction process to the end. The following lemma can be verified by induction.

LEMMA 3.3. Let  $S_\ell$  and  $\tilde{S}_\ell$  be recursively defined by

$$\begin{aligned} S_0 &= 0, & \tilde{S}_0 &= 0, \\ S_1 &= H_b, & \tilde{S}_1 &= H_{b(1:k,1:k)}, \\ S_\ell &= H_a S_{\ell-1} + H_b S_{\ell-2} \tilde{S}_\ell = H_{a(1:k,1:k)} \tilde{S}_{\ell-1} + H_{b(1:k,1:k)} \tilde{S}_{\ell-2} \end{aligned}$$

for  $\ell \geq 2$ . Then

$$L^\ell = \begin{pmatrix} S_{\ell-1} & \mathbf{x} \\ S_\ell & \mathbf{x} \end{pmatrix} \quad \text{and} \quad L_k^\ell = \begin{pmatrix} \tilde{S}_{\ell-1} & \mathbf{x} \\ \tilde{S}_\ell & \mathbf{x} \end{pmatrix}.$$

As  $H_a$  and  $H_b$  are banded with lower bandwidth  $p + 1$ , it is clear that  $S_\ell$  and  $\tilde{S}_\ell$  are also banded but with lower bandwidth  $\ell(p + 1)$ .

LEMMA 3.4. Suppose  $k \geq 3$ , and let  $m = \lfloor \frac{k}{p+1} \rfloor$  (the largest integer  $\leq \frac{k}{p+1}$ ). Then

1.  $S_\ell e_1 = \binom{k}{n-k} \binom{\tilde{S}_\ell e_1}{0}$  for  $\ell = 0, 1, \dots, m$ ,
2.  $S_{m+1} e_1 = \binom{k}{n-k} \binom{\tilde{S}_{m+1} e_1}{\mathbf{x}}$ .

*Proof.* We shall prove claim 1 by induction on  $\ell$ . It holds true for  $\ell = 0, 1$ . Suppose  $m \geq \ell \geq 2$  and that the claim holds for  $0, 1, \dots, \ell - 1$ . Then  $\ell(p + 1) \leq k$  and

$$\begin{aligned} S_\ell e_1 &= H_a S_{\ell-1} e_1 + H_b S_{\ell-2} e_1 \\ &= H_a \begin{pmatrix} \tilde{S}_{\ell-1} e_1 \\ 0 \end{pmatrix} + H_b \begin{pmatrix} \tilde{S}_{\ell-2} e_1 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} H_{a(1:k,1:k)} \tilde{S}_{\ell-1} e_1 \\ 0 \end{pmatrix} + \begin{pmatrix} H_{b(1:k,1:k)} \tilde{S}_{\ell-2} e_1 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} \tilde{S}_\ell e_1 \\ 0 \end{pmatrix}, \end{aligned}$$

where we note that  $\tilde{S}_{\ell-1} e_1$  and  $\tilde{S}_{\ell-2} e_1$  have at most the first  $(\ell - 1)(p + 1)$  entries nonzero and  $H_a$  and  $H_b$  have lower bandwidth  $p + 1$ . Claim 1 is therefore proved. With claim 1 proved, setting  $\ell = m + 1$  in the above equations leads to claim 2.  $\square$

It follows from the above lemma that  $e_1^* S_\ell e_1 = e_1^* \tilde{S}_\ell e_1$  for  $\ell = 0, 1, \dots, m + 1$  ( $m = \lfloor \frac{k}{p+1} \rfloor$ ). (Recall that  $e_1$  is the first column of  $I$  of appropriate dimension.) Then,

$$e_1^* L^{\ell+1} e_1 = e_1^* L_k^{\ell+1} e_1.$$

Therefore, for any polynomial  $f$  of degree  $m + 2$ ,

$$(3.4) \quad e_1^* f(L) e_1 = e_1^* f(L_k) e_1.$$

We now derive from this equation some relations between the eigenvalues of  $L$  and  $L_k$ . For the sake of simplicity, we assume that  $L$  and  $L_k$  are diagonalizable and write

$$(3.5) \quad L_k = U^* \Theta V \quad \text{and} \quad L = X^* \Lambda Y,$$

where  $\Theta = \text{diag}(\theta_1, \dots, \theta_{2k})$ ,  $U^* V = I$ , and  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_{2n})$ ,  $X^* Y = I$ . Write  $U = (u_{ij})$ ,  $V = (v_{ij})$ ,  $X = (x_{ij})$ , and  $Y = (y_{ij})$ . Substituting (3.5) into (3.4), we obtain

$$e_1^* X^* f(\Lambda) Y e_1 = e_1^* U^* f(\Theta) V e_1.$$

Thus

$$\sum_{i=1}^{2n} f(\lambda_i) \bar{x}_{i1} y_{i1} = \sum_{i=1}^{2k} f(\theta_i) \bar{u}_{i1} v_{i1}.$$

Without loss of generality, we consider approximation of  $\lambda_1$  and assume that  $|\lambda_1 - \theta_1| = \min_j |\lambda_1 - \theta_j|$ . Then, for any polynomial  $p$  of degree  $m + 1$ , we use  $f(t) = (t - \theta_1)p(t)$  in the above and obtain

$$\lambda_1 - \theta_1 = \frac{1}{p(\lambda_1) \bar{x}_{11} y_{11}} \left[ - \sum_{i=2}^{2n} (\lambda_i - \theta_1) p(\lambda_i) \bar{x}_{i1} y_{i1} + \sum_{i=2}^{2k} (\theta_i - \theta_1) p(\theta_i) \bar{u}_{i1} v_{i1} \right].$$

Bounding  $p(\lambda_i)$ ,  $p(\theta_i)$  by their maximum, we obtain

$$|\lambda_1 - \theta_1| \leq \frac{\max_{i \neq 1} \{|p(\lambda_i)|, |p(\theta_i)|\} \sum_{i=2}^{2n} |(\lambda_i - \theta_1) \bar{x}_{i1} y_{i1}| + \sum_{i=2}^{2k} |(\theta_i - \theta_1) \bar{u}_{i1} v_{i1}|}{|p(\lambda_1)| |\bar{x}_{11} y_{11}|},$$

which leads to the following theorem.

**THEOREM 3.5.** *Let  $|\lambda_1 - \theta_1| = \min_j |\lambda_1 - \theta_j|$ . Then we have*

$$|\lambda_1 - \theta_1| \leq K \epsilon_{m+1} \frac{\sqrt{\sum_{i \neq 1} (|x_{i1}|^2 + |u_{i1}|^2)}}{|x_{11}|} \cdot \frac{\sqrt{\sum_{i \neq 1} (|y_{i1}|^2 + |v_{i1}|^2)}}{|y_{11}|},$$

where

$$\epsilon_\ell = \min_{\deg p = \ell, p(\lambda_1) = 1} \max_{i \neq 1} \{|p(\lambda_i)|, |p(\theta_i)|\},$$

$m = \lfloor \frac{k}{p+1} \rfloor$ , and  $K = \max_{i \neq 1} \{|\lambda_i - \theta_1|; |\theta_i - \theta_1|\}$ .

$\epsilon_{m+1}$  is the dominating factor in the bound and can be bounded with the Chebyshev polynomials under some assumptions of the eigenvalue distribution (see [29, p. 191] for details). Essentially, if  $\lambda_1$  and  $\theta_1$  are well separated from the other  $\lambda_i$  and  $\theta_i$ , then  $\epsilon_{m+1}$  can be made small and the bound shows that a good approximation of  $\lambda_1$  is expected. The last two factors in the bound are related to the angle between  $q_1$  and the right and left eigenvectors corresponding to  $\lambda_1$  and show the dependence of convergence on the initial vector.

**3.2. Shift-and-invert transform.** The Arnoldi-type algorithm is often combined with a shift-and-invert transformation to accelerate convergence [8]. For example, to compute the eigenvalues near  $\lambda_0$ , a transformation of the form  $\mu = (\lambda - \lambda_0)^{-1}$  is usually used, but this would destroy the low rank perturbation property. It turns out that the transformation

$$(3.6) \quad 1/\lambda = 1/\mu + 1/\lambda_0$$

also maps the eigenvalues  $\lambda$  close to  $\lambda_0$  to large and well-separated  $\mu$ , and more importantly it preserves the low rank perturbation property. Indeed,

$$\begin{aligned} \lambda^2 I - \lambda A - B &= \lambda^2 (I - (1/\lambda)A - (1/\lambda)^2 B) \\ &= \lambda^2 [I - (1/\mu + 1/\lambda_0)A - (1/\mu + 1/\lambda_0)^2 B] \\ &= \lambda^2 [I - (1/\lambda_0)A - (1/\lambda_0)^2 B - (1/\mu)(A + 2/\lambda_0 B) - (1/\mu)^2 B] \\ (3.7) \quad &= (\lambda/\mu)^2 M(\mu^2 I - \mu \hat{A} - \hat{B}), \end{aligned}$$

where

$$\begin{aligned} M &= I - (1/\lambda_0)A - (1/\lambda_0)^2B, \\ \hat{A} &= M^{-1}(A + 2/\lambda_0B), \\ \hat{B} &= M^{-1}B. \end{aligned}$$

For  $\zeta B + \xi A = E$ , we have

$$(\zeta - 2\xi/\lambda_0)\hat{B} + \xi\hat{A} = M^{-1}E,$$

which is still of low rank.

**4. A constrained least squares problem.** Let<sup>2</sup>  $H \in \mathbb{R}^{n \times n}$  be symmetric and  $g \in \mathbb{R}^n$ . We consider the constrained minimization problem

$$(4.1) \quad \min_{x^T x = \delta^2} x^T H x - 2g^T x.$$

As pointed out in the introduction, this problem arises in the regularization of discretized ill-posed problems and trust-region subproblems. The Lagrangian equations for (4.1) are

$$(4.2) \quad Hx - g = \lambda x,$$

$$(4.3) \quad x^T x = \delta^2,$$

where  $\lambda$  is the Lagrangian multiplier. It is shown in Gander [10] that the solution  $(\lambda, x)$  to the Lagrange equation (4.2), (4.3) with the smallest  $\lambda$  solves (4.1). Furthermore, it is shown by Gander, Golub, and von Matt [11] that (4.2) and (4.3) can be reduced to the QEP

$$(4.4) \quad (\lambda^2 I - 2\lambda H + H^2 - \delta^{-2} g g^T) y = 0.$$

Specifically, it is proved that if  $(\lambda, x)$  solves (4.2) and (4.3), then  $\lambda$  is an eigenvalue of (4.4). Conversely, for an eigenpair  $(\lambda, y)$  of (4.4), if  $\lambda \notin \lambda(H)$ , then  $(\lambda, x)$  with  $x = (H - \lambda I)^{-1} g$  solves (4.2) and (4.3); if  $\lambda \in \lambda(H)$ , then  $\lambda$  is a solution to (4.2) and (4.3) if and only if  $x = (H - \lambda I)^\dagger g$  satisfies  $(H - \lambda I)x = g$  and  $x^T x \leq \delta^2$ , where  $(H - \lambda I)^\dagger$  is the pseudo-inverse [7, 14].

For small problems, it appears that the solution through (4.4) is not competitive when compared with other direct methods; see [11]. For large scale problems, however, we will show that (4.4) can be solved efficiently by the Arnoldi-type process, and thus it offers a very promising approach to solving (4.1).

In the setting of large scale problems, the eigenvalue problem (4.4) is usually solved only approximately by an iterative method that reduces the residual of the approximate solution to certain threshold. Here we first consider when an approximate solution of (4.4) leads to an approximate solution of (4.2) and (4.3). The following theorem is an inexact version of the result presented in [11] and reveals an interesting numerical issue associated with using (4.4).

**THEOREM 4.1.** *Let  $(\theta, y)$  with  $\|y\| = 1$  be an approximate eigenpair of (4.4) and let*

$$(4.5) \quad r = (\theta^2 I - 2\theta H + H^2 - \delta^{-2} g g^T) y.$$

*Assume  $g^T y \neq 0$ .*

---

<sup>2</sup>We restrict our discussion in this section to real matrices so as to be consistent with existing related literature. Obviously the section can be extended to cover the complex case in which  $H$  is Hermitian.

1. Let  $z = \frac{\delta^2}{g^T y}(H - \theta I)y$ . We have

$$(4.6) \quad (H - \theta I)z - g = \frac{\delta^2}{g^T y}r,$$

$$(4.7) \quad \frac{z^T z - \delta^2}{\delta^2} = \frac{\delta^2}{(g^T y)^2}y^T r.$$

In particular, if  $y^T r = 0$  (which is the case if  $(\theta, y)$  is obtained from the Arnoldi-type process), then  $z^T z - \delta^2 = 0$  by (4.7).

2. If  $\theta \notin \lambda(H)$ , let  $\hat{z} = (H - \theta I)^{-1}g$ . We have

$$(H - \theta I)\hat{z} - g = 0, \\ \frac{\hat{z}^T \hat{z} - \delta^2}{\delta^2} = \frac{\hat{z}^T (H - \theta I)^{-1}r}{g^T y}.$$

*Proof.* From (4.5), it follows that

$$(H - \theta I)^2 y = \delta^{-2} g g^T y + r,$$

which implies  $(H - \theta I)z - g = \frac{\delta^2}{g^T y}r$ . Using the definition of  $z$ , we have

$$z^T z = \frac{\delta^4}{(g^T y)^2} y^T (H - \theta I)^2 y \\ = \frac{\delta^4}{(g^T y)^2} y^T (\delta^{-2} g g^T y + r) \\ = \delta^2 + \frac{\delta^4}{(g^T y)^2} y^T r.$$

This proves (4.7). For part 2,  $(H - \theta I)\hat{z} - g = 0$  follows directly from the definition of  $\hat{z}$ . Furthermore, from (4.5),  $\frac{g^T y}{\delta^2}(H - \theta I)^{-2}g = y - (H - \theta I)^{-2}r$ . Thus

$$\hat{z}^T \hat{z} = g^T (H - \theta I)^{-2}g \\ = \frac{\delta^2}{g^T y} (g^T y - g^T (H - \theta I)^{-2}r) \\ = \delta^2 - \frac{\delta^2}{g^T y} \hat{z} (H - \theta I)^{-1}r,$$

which leads to the second equation.  $\square$

Once an approximation to the smallest eigenpair is found, then either  $x \approx z$  or  $x \approx \hat{z}$  gives an approximate solution to (4.1). However,  $\hat{z}$  requires solving  $(H - \theta I)\hat{z} = g$ , and the constraint error  $(\hat{z}^T \hat{z} - \delta^2)/\delta^2$  can be large. On the other hand, taking  $x \approx z$  is more straightforward. We will consider  $z = \frac{\delta^2}{g^T y}(H - \theta I)y$  only.

The theorem illustrates a potential difficulty to construct a solution of (4.2) and (4.3) from an approximate eigenpair. The error for the constraint equation (4.3) is inversely proportional to  $(g^T y)^2$  and, in discretized ill-posed problems,  $g^T y$  is typically very small. Thus, an approximate eigenpair with small residual  $r$  does not necessarily lead to a good approximate solution to the Lagrange equations. Fortunately, the theorem also shows that this problem is eliminated if we have  $y^T r = 0$ . For  $(\theta, y)$  as obtained from the Arnoldi-type algorithm, we have  $y^T r = 0$  since  $r \perp \text{span}\{Q_{(:,1:k)}\}$

and  $y \in \text{span}\{Q_{(:,1:k)}\}$  (see Theorem 3.2). Hence  $z$  will always satisfy the constraint, but this is valid in theory only. In practice, we have only near orthogonality between  $y$  and  $r$ , but this orthogonality can be further improved by recomputing  $\theta$  to enforce orthogonality  $y^T r = 0$ . Namely, if  $(\theta, y)$  is an approximate eigenpair, we recompute  $\theta$  as the Rayleigh quotient by solving

$$(4.8) \quad \theta^2 I - 2\theta y^T H y + y^T (H^2 - \delta^{-2} g g^T) y = 0.$$

This will lead to much improved orthogonality  $y^T r = 0$  and will hence keep the error in the constraint equation small (see examples in section 5.2). The importance of the orthogonality  $y^T r = 0$  can be highlighted by considering the QR algorithm. If  $(\theta, y)$  is obtained from the QR algorithm, we know  $r \approx \mathcal{O}(\epsilon)$  but cannot say anything about the direction of  $r$ , which implies  $y^T r$  is of order  $\epsilon$  only. Using  $(\theta, y)$  directly to compute  $z$ , the error in the constraint equation (4.7) can be very large, even when  $g^T y$  is modestly small (e.g., of order  $\sqrt{\epsilon}$ ); see [11] for some numerical results. This problem can be corrected by recomputing  $\theta$  through (4.8) to enforce the orthogonality.

The theorem is valid only when  $g^T y \neq 0$ . If  $g^T y = 0$  and  $y$  is an exact eigenvector (i.e.,  $r = 0$ ), then  $(H - \theta I)^2 y = 0$ . Since  $H - \theta I$  is real symmetric, we have  $(H - \theta I)y = 0$ , and hence  $\theta$  is an eigenvalue of  $H$  with  $y$  a corresponding eigenvector. In this case,  $\theta$  is a solution to the Lagrange equation if and only if  $x = (H - \theta I)^\dagger g$  satisfies  $(H - \theta I)x = g$  and  $x^T x \leq \delta^2$ . This is indeed an extreme situation called *the hard case* of (4.1) (see [23]). In the hard case, the solution does not depend continuously on  $g$ .

We now show that the QEP (4.4) can be efficiently solved by the Arnoldi-type algorithm. While theoretically we can apply the Arnoldi-type process directly to  $H$  and  $H^2 - \delta^{-2} g g^T$ , it is easier to do it indirectly by using Algorithm 2.1 on  $H$  and  $g g^T$  first, from which a reduction of  $H^2 - \delta^{-2} g g^T$  can be derived.

Let Algorithm 2.1 (or the symmetric version) be applied to  $A = H$  and  $B = g g^T$  for  $k$  steps; we obtain

$$\begin{aligned} A Q_{(:,1:k)} &= Q_{(:,1:k+2)} H_a (1:k+2, 1:k), \\ B Q_{(:,1:k)} &= Q_{(:,1:k+2)} H_b (1:k+2, 1:k). \end{aligned}$$

Since  $A$  and  $B$  are symmetric and  $B$  is of rank 1,  $H_a$  and  $H_b$  are symmetric banded with bandwidth 2. Indeed,  $B q_1 - q_1 h_{b;11} - q_2 h_{b;21} = q_3 h_{b;31}$ , i.e.,  $g(g^T q_1) = q_1 h_{b;11} + q_2 h_{b;21} + q_3 h_{b;31} = Q_{(:,1:3)} H_b (1:3, 1)$ . Then

$$\begin{aligned} B Q_{(:,1:k)} &= g g^T Q_{(:,1:k)} = \frac{1}{(g^T q_1)^2} Q_{(:,1:3)} (H_b (1:3, 1) H_b^T (1:3, 1)) Q_{(:,1:3)}^T Q_{(:,1:k)} \\ &= \frac{1}{(g^T q_1)^2} Q_{(:,1:3)} [H_b (1:3, 1) H_b^T (1:3, 1), 0]. \end{aligned}$$

Thus,

$$H_b (1:k+2, 1:k) = \begin{pmatrix} H_b (1:3, 1) H_b^T (1:3, 1) / (g^T q_1)^2 & 0 \\ 0 & 0 \end{pmatrix}.$$

In fact, with  $H_b$  as defined above, the algorithm can be implemented with the  $B$  part (line 13 to line 20 of Algorithm 2.1) omitted after  $j > 2$ . Furthermore,

$$A^2 Q_{(:,1:k)} = A Q_{(:,1:k+2)} H_a (1:k+2, 1:k) = Q_{(:,1:k+4)} H_a (1:k+4, 1:k+2) H_a (1:k+2, 1:k).$$

Thus

$$(\delta^{-2} g g^T - H^2) Q_{(:,1:k)} = Q_{(:,1:k+4)} \hat{H}_b (1:k+4, 1:k),$$

where  $\hat{H}_b = \delta^{-2}H_b - H_a^2$ . Clearly  $\hat{H}_b$  has a bandwidth 4. We can now approximate (4.1) by solving the reduced problem

$$\theta^2 I - 2\theta H_{a(1:k,1:k)} - \hat{H}_{b(1:k,1:k)},$$

where

$$\hat{H}_{b(1:k,1:k)} = \delta^{-2}H_{b(1:k,1:k)} - H_{a(1:k+2,1:k)}^T H_{a(1:k+2,1:k)}.$$

Noting that  $A$  and  $B$  are symmetric, we can use the symmetric version of Algorithm 2.1 here. We observe that the approximate eigenpair  $(\theta_k, y_k)$  as obtained from this algorithm still satisfies the Galerkin-type condition (3.3). We summarize the process into the following algorithm for solving (4.1).

ALGORITHM 4.1 (Lanczos-type process for constrained minimization problem).

1. Input:  $H$ ,  $g$ , and  $q_1$  with  $\|q_1\|_2 = 1$ ;
2.  $\hat{q} = Hq_1$ ;
3.  $h_{a;11} = q_1^T \hat{q}$ ;  $\hat{q} = \hat{q} - q_1 h_{a;11}$ ;
4.  $h_{a;21} = \|\hat{q}\|_2$ ;  $q_2 = \hat{q}/h_{a;21}$ ;
5.  $h_{b;11} = (g^T q_1)^2$ ;  $h_{b;21} = q_2^T g(g^T q_1)$ ;
6.  $\hat{q} = g(g^T q_1) - q_1 h_{b;11} - q_2 h_{b;21}$ ;
7.  $h_{b;31} = \|\hat{q}\|_2$ ;  $q_3 = \hat{q}/h_{b;31}$ ;
8.  $N = 3$
9. For  $j = 2, \dots, k$
10.  $\hat{q} = Hq_j$ ;
11. For  $i = \max\{1, j-2\} : N$  do
12.  $h_{a;ij} = q_i^T \hat{q}$ ;  $\hat{q} = \hat{q} - q_i h_{a;ij}$ ;
13. EndDo
14.  $h_{a;N+1,j} = \|\hat{q}\|_2$ ;
15. If  $h_{a;N+1,j} > 0$ ,
16.  $N = N + 1$ ,  $q_N = \hat{q}/h_{a;Nj}$ ;
17. EndIf;
18. If  $N \leq j$ , break;
19. EndDo
20.  $H_{b(1:k,1:k)} = \begin{pmatrix} H_{b(1:3,1)} H_{b(1:3,1)}^T / (g^T q_1)^2 & 0 \\ 0 & 0 \end{pmatrix}$ ;
21.  $\hat{H}_{b(1:k,1:k)} = \delta^{-2}H_{b(1:k,1:k)} - H_{a(1:k+2,1:k)}^T H_{a(1:k+2,1:k)}$ ;
22. Find the smallest real eigenpair  $(\theta_k, v_k)$  of  $I\theta^2 - 2H_{a(1:k,1:k)}\theta - \hat{H}_{b(1:k,1:k)}$ ;
23.  $y_k = Q_{(:,1:k)} v_k$ ;
24.  $\theta$  is the root of  $\theta^2 I - 2\theta y_k^T H y_k + \|H y_k\|^2 - \delta^{-2}(y_k^T g)^2 = 0$  that is closer to  $\theta_k$ ;
25.  $z_k = \frac{\delta^2}{g^T y_k} (H - \theta I) y_k$ .

In the algorithm, the iteration number  $k$  can be determined by requiring that the solution  $z_k$  satisfies, for example,  $\|H z_k - \theta_k z_k - g\|/\|g\| \leq \text{tol}_1$  and  $|z_k^T z_k - \delta^2|/\delta^2 \leq \text{tol}_2$  for some given tolerances  $\text{tol}_1$  and  $\text{tol}_2$ .

Finally, we note that with its special structure, the QEP (4.4) can also be solved by using the standard Lanczos algorithm; namely, we can apply  $k$  step of the Lanczos algorithm to an initial vector  $q_1$  to produce  $Q_{k+1} = [q_1, q_2, \dots, q_k, q_{k+1}]$  with orthonormal columns such that

$$AQ_k = Q_{k+1} T_{(1:k+1,1:k)},$$

where  $T$  is  $n \times n$  tridiagonal. Then, if  $h = Q_k^T g$ , we can approximate (4.4) by its projection  $Q_k^T(\lambda^2 I - 2\lambda H + H^2 - \delta^{-2} g g^T) Q_k$ , which is

$$(4.9) \quad \lambda^2 I - 2\lambda T_{(1:k,1:k)} + T_{(1:k+1,1:k)}^T T_{(1:k+1,1:k)} - \delta^{-2} h h^T.$$

In this case, the choice of  $q_1$  plays an important role, as we need the Krylov subspace  $\text{span}\{Q_k\}$  to approximate well both  $g$  and the eigenvector sought. If  $q_1$  is chosen to be a random vector,  $g$  may not be well approximated by its projection onto  $\text{span}\{Q_k\}$ . On the other hand, if  $q_1 = g/\|g\|$ , then  $g \in \text{span}\{Q_k\}$  but the eigenvector sought is not necessarily well approximated by  $\text{span}\{Q_k\}$ . We note that the choice of  $g$  works out quite well compared to our process with a random  $q_1$  for discrete ill-posed problems that we tested.

**5. Numerical examples.** In this section we shall present two sets of numerical examples. In the first set, we use random sparse matrices as generated by MATLAB. The second set is for the constrained least squares problems (4.1) as arising in the regularization solution of discretized ill-posed problems [23].

**5.1. QEP with random matrices.** We start by testing on QEP  $\lambda^2 I - \lambda A - B$  with no relation between  $A$  and  $B$  assumed, where  $A$  and  $B$  are generated by MATLAB commands

$$n = 500; \quad A = \text{sprandn}(n, n, 0.05); \quad B = \text{sprandn}(n, n, 0.05);$$

initial vector  $q_1$  is a random vector. A direct application of Krylov-type methods to random matrices gives poor convergence results. Instead, we use a shift-and-invert transformation with the shift  $\lambda_0 = -1.0 + 3i$ , which gives a much more favorable spectral distribution. Then applying Algorithm 2.1 with  $k = 8$  on the transformed problems as in (3.6) and (3.7), an approximate eigenvalue  $\lambda_1 \approx -0.9549 + 2.8519i$  is computed. Figure 1 plots the normalized residual

$$(5.1) \quad \gamma_j \equiv \frac{\|(\lambda_j^2 I - A\lambda_j - B)x_j\|}{\max\{|\lambda_j|^2 \|x_j\|, |\lambda_j| \|Ax_j\|, \|Bx_j\|\}}$$

for all eigenvalues obtained, where  $\lambda_j$  is a computed eigenvalue and  $x_j$  is a corresponding computed eigenvector. Notice that since both  $A$  and  $B$  are randomly generated and thus unrelated, every application of  $A$  or  $B$  on  $q$ -vectors produces new directions, and consequently  $N = 2k + 1 = 17$  and there are 34 approximate eigenvalues.

Next we test Algorithm 2.1 on the low rank cases. The matrices  $A$  and  $B$  are generated as

$$n=500; \quad A=\text{sprandn}(n,n,0.05); \\ X=\text{randn}(n,2); \quad Y=\text{randn}(n,2); \quad B=1.1*A+2.3*X*Y'$$

Thus  $-1.1A + B = 2.3XY'$ , of rank 2. But in running Algorithm 2.1, we do not assume knowing  $X$  and  $Y$ . Without shifting and with a random  $q_1$  and  $k = 30$ , Algorithm 2.1 outputs  $N = 33$  and  $H_{a(1:N,1:N)}$  and  $H_{b(1:N,1:N)}$ . Figure 2 plots the residual errors for the 66 Ritz values obtained, where computed  $\lambda_{51}, \lambda_{52} = -1.1345 \pm 0.0307i$  and  $\lambda_{65}, \lambda_{66} = -1.0561 \pm 0.0168i$ . The sparsity patterns  $H_{a(1:N,1:N)}$  and  $H_{b(1:N,1:N)}$  are displayed in Figure 3.

Now we apply the shift-and-invert transformation of (3.6), which will preserve the low rank perturbation property (see section 3.2). We take  $\lambda_0 = -1.2 + i$  and apply Algorithm 2.1 with  $k = 15$  and an random  $q_1$  on the transformed problems as in (3.6) and (3.7). Figure 4 plots the residual errors of the computed approximate

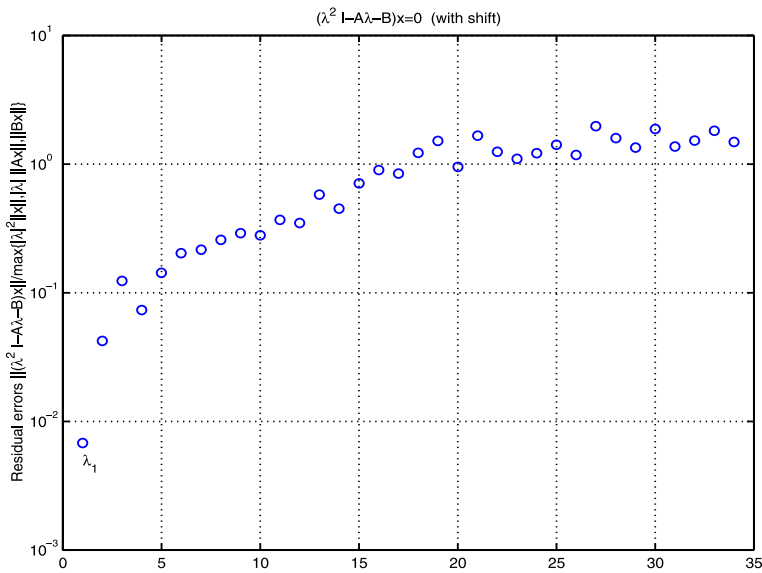


FIG. 1. Residual errors of computed eigenvalues:  $A$  and  $B$  unrelated.

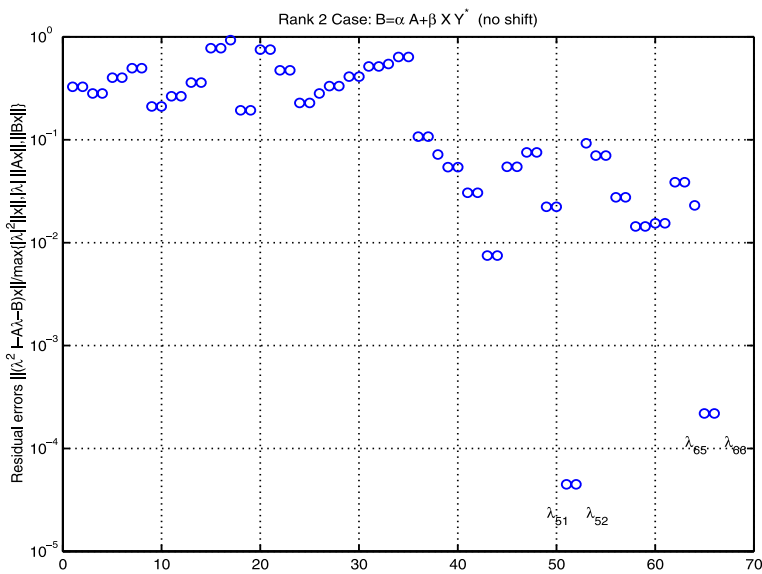


FIG. 2. Residual errors of computed eigenvalues: A rank 2 case.

eigenvalues, where computed  $\lambda_1 = -1.1415 + 0.9082i$  and  $\lambda_{35} = -1.1725 - 1.3274i$ . With  $N = 18$ , the projections have the same sparsity structure as in Figure 3, while the convergence is clearly accelerated.

**5.2. Constrained least squares problems.** We now consider some constraint least squares testing problems (1.3) taken from the regularization tool of Hansen [15]. They are discretizations of some integral equations (see [15] for more detailed

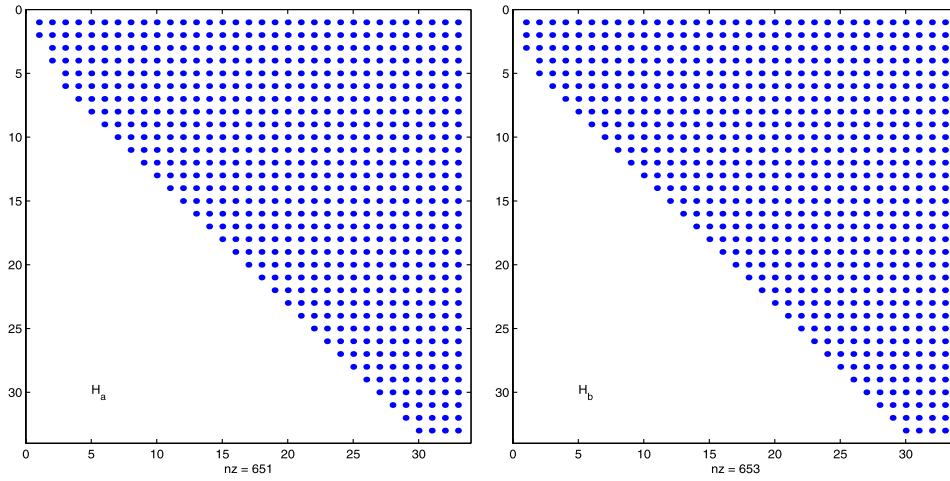


FIG. 3. Sparsity patterns of  $H_a$  and  $H_b$ : A rank 2 case.

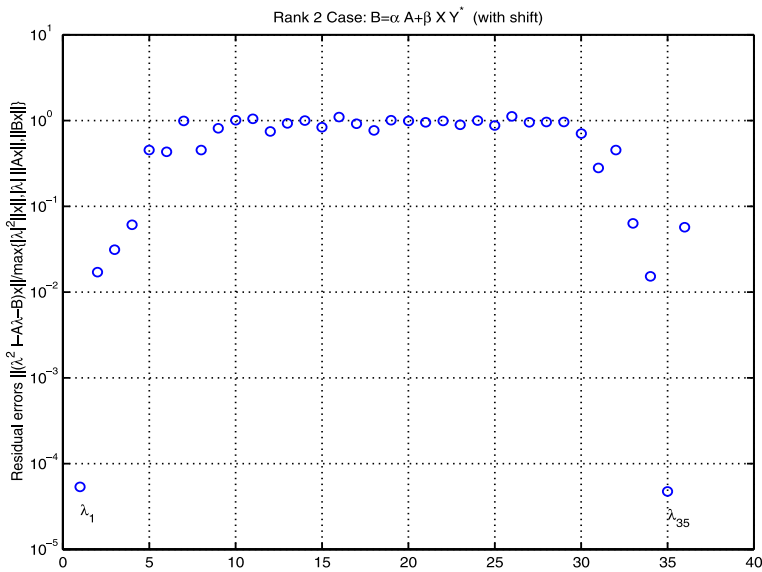


FIG. 4. Residual errors of computed eigenvalues: A rank 2 case with shift.

description of the matrices). In all test problems except `parallax` and `ursell`, a reference solution  $x_{IP}$  is provided by the routine and, in that case, we set  $\delta = \|x_{IP}\|$ . We also set the dimension  $n = 1000$  for all tests except for `blur` (image deblurring problem) for which  $n = 32^2$  due to the problem's characteristic. Typically, the matrix  $H$  is either of low rank (with a rectangular  $C$ ) or numerically of low rank (with a large number of tiny singular values). This appears to be one of the reasons for very fast convergence that we will see.

We first test the convergence of the eigenvalue with the smallest real part. Here we use a random vector as the initial vector and terminate the iteration when the

TABLE 1  
*QEP from constraint least squares problems ( $\gamma_k$ —normalized residual).*

Problem	$\theta_k$	$\gamma_k$	$ \theta_k - \lambda_{\text{QR}} $	$k$
<b>barrt</b>	$4.66188e - 08$	$2.1e - 13$	$4.8e - 08$	4
<b>ill heat</b>	$7.47851e - 08$	$3.0e - 09$	$7.6e - 08$	18
<b>well heat</b>	$1.19617e - 08$	$9.9e - 09$	$1.3e - 08$	188
<b>blur</b>	$1.34996e - 12$	$9.3e - 09$	$1.3e - 12$	347
<b>deriv2 (1)</b>	$4.42695e - 08$	$3.5e - 09$	$4.5e - 08$	8
<b>deriv2 (2)</b>	$4.59196e - 08$	$2.5e - 09$	$4.6e - 08$	8
<b>deriv2 (3)</b>	$6.68204e - 08$	$2.9e - 09$	$6.7e - 08$	7
<b>foxgood</b>	$2.22965e - 09$	$5.2e - 13$	$6.9e - 09$	3
<b>parallax</b>	$-1.34982e - 01$	$9.1e - 10$	$1.3e - 15$	10
<b>phillips</b>	$3.17750e - 05$	$1.7e - 09$	$5.0e - 05$	11
<b>shaw</b>	$1.01446e - 04$	$3.2e - 09$	$1.0e - 04$	6
<b>spikes</b>	$1.64716e - 02$	$7.1e - 09$	$1.6e - 02$	10
<b>ursell</b>	$-2.42031e - 01$	$1.0e - 12$	$3.1e - 16$	4
<b>wing</b>	$1.13499e - 07$	$3.4e - 11$	$1.1e - 07$	3

normalized residual (5.1) satisfies  $\gamma_k < 10^{-8}$ . Table 1 lists the results obtained, where we include the computed Ritz value  $\theta_k$ , the normalized residual  $\gamma_k$ , the errors  $|\theta_k - \lambda_{\text{QR}}|$  ( $\lambda_{\text{QR}}$  is the leftmost eigenvalue returned by the QR algorithm (`eig` of MATLAB) on  $A_{\text{LIN}}$ ), and the required number of iterations  $k$ . We note that for those problems where  $Cx_{\text{IP}} \approx b$  (cf. (1.3)),  $x_{\text{IP}}$  is a solution to (4.1) because it satisfies the constraint. Then,  $Hx_{\text{IP}} - g \approx 0$ , and therefore the eigenvalue is nearly 0 for those problems.

In all problems, the residual falls below the given threshold within a small number of iterations. For the problems where the smallest eigenvalue is 0 or nearly 0, the absolute error of eigenvalue is approximately equal to  $\theta$  and is approximately  $10^{-5}$  or smaller except for the **spike** problem. For the other problems (**parallax** and **ursell**, in which  $x_{\text{IP}}$  is not given and  $\delta = \|b\|$ ), eigenvalues are of  $\mathcal{O}(1)$ , and the absolute errors are then of  $\mathcal{O}(10^{-15})$  as compared with the QR algorithm. For the **spike** problem, the large eigenvalue error is due to the fact that the norm of  $H^2$  is so large ( $\|H^2\|_1 \approx \mathcal{O}(10^{10})$ ) that the absolute residual  $\|r_k\|$  is only reduced to  $\mathcal{O}(1)$ .

In Figure 5, we present the residual convergence history for the inverse heat problem (ill-conditioned **heat** with  $\kappa = 1$ ). The solid line is for the normalized residual  $\gamma_k$  (5.1) and the dotted line for the error  $|\theta_k - \lambda_{\text{QR}}|$ .

We also present a comparison among Algorithm 4.1, (4.9) with  $q_1 = g$ , and (4.9) with random  $q_1$  that directly use the projection onto the Krylov subspace generated by  $H$  and  $q_1$ . Figure 6 compares convergence history of normalized residuals for the **heat** problem. It appears that with the choice  $q_1 = g$ , the direct approach (4.9) and Algorithm 4.1 have a very similar convergence characteristic, with the former converging a few steps faster and the latter being slightly more stable after the residual has converged to the level of machine precision. The random choice of  $q_1$ , on the other hand, can result in slower convergence, as expected.

We next test convergence of the approximate solution  $z_k$  to the Lagrange equations. Here we terminate the iteration whenever both the relative residual and the constraint error are below  $10^{-6}$ , i.e., when

$$\zeta_k \equiv \frac{\|Hz_k - \theta_k z_k - g\|}{\|g\|} < 10^{-6} \quad \text{and} \quad \eta_k \equiv \frac{z_k^T z_k - \delta^2}{\delta^2} < 10^{-6}.$$

In addition to the residual  $\zeta_k$ , the constraint error  $\eta_k$ , the relative error  $\|z_k - x_{\text{IP}}\|/\|x_{\text{IP}}\|$

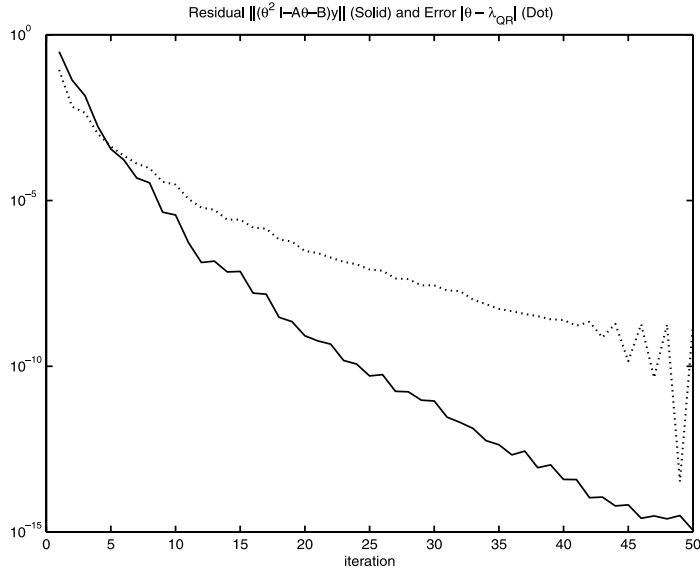


FIG. 5. Eigenvalue convergence history for the heat problem with  $\kappa = 1$ .

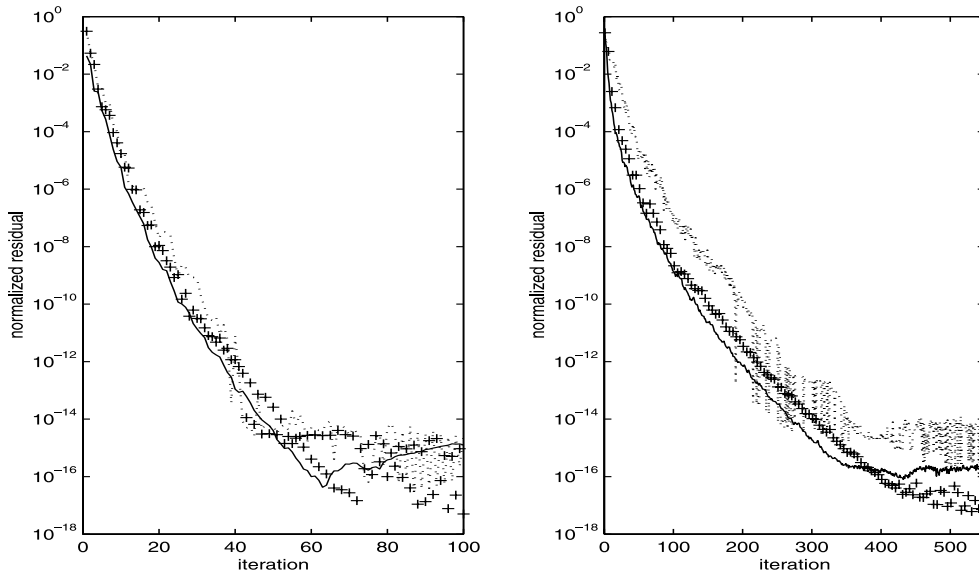


FIG. 6. Comparisons on the heat problem. Left:  $\kappa = 1$ . Right:  $\kappa = 3$ . Here  $+line$ : Algorithm 4.1;  $dashed$ : (4.9) with  $q_1 = g$ ;  $dotted$ : (4.9) with random  $q_1$ .

(where  $x_{IP}$  is available), and iteration number  $k$ , Table 2 also displays  $\|r_k\|$ ,  $|y_k^T r_k|$ , and  $|g^T y_k|$ , which relates  $\eta_k$  and  $\zeta_k$  to the eigenvalue residual  $\|r_k\|$  (see Theorem 4.1).

Figure 7 plots the convergence history of  $z_k$  for the inverse heat problem.

These numerical results show that a solution to the Lagrange equations to the desired accuracy is obtained within a small number of iterations  $k$  for all but the spike problem. The speed of convergence compares favorably with that of the LSTRS

TABLE 2  
 $\eta_k = \frac{z_k^T z_k - \delta^2}{\delta^2}$ ,  $\zeta_k = \frac{\|Hz_k - \theta_k z_k - g\|}{\|g\|}$ .

Problem	$\delta$	$\eta_k$	$\gamma_k$	$\frac{\ z_k - x_{IP}\ }{\ x_{IP}\ }$	$k$	$\ r_k\ $	$ y_k^T r_k $	$ g^T y_k $
barrt	1.2	$2e-12$	$3e-7$	$1e-1$	4	$3e-08$	$6e-17$	$1e-2$
ill heat	7.7	$9e-16$	$4e-7$	$2e-2$	26	$5e-13$	$3e-23$	$1e-4$
well heat	7.7	$2e-15$	$9e-7$	$4e-4$	112	$6e-08$	$5e-17$	$1e+0$
blur	36.5	$1e-15$	$9e-7$	$1e-5$	327	$1e-08$	$6e-19$	$7e-1$
deriv2 (1)	0.6	$1e-16$	$8e-7$	$2e-1$	19	$1e-15$	$1e-26$	$1e-7$
deriv2 (2)	1.7	$2e-16$	$9e-7$	$1e-1$	19	$1e-15$	$1e-26$	$3e-7$
deriv2 (3)	0.3	$5e-16$	$2e-7$	$9e-3$	10	$4e-14$	$8e-24$	$6e-6$
foxgood	18.2	$8e-16$	$1e-8$	$7e-3$	4	$1e-11$	$3e-20$	$2e-2$
parallax	18.1	$1e-16$	$2e-8$	—	13	$9e-15$	$1e-23$	$3e-05$
phillips	2.9	$1e-15$	$5e-7$	$8e-3$	11	$1e-06$	$1e-16$	$2e-1$
shaw	31.5	$1e-15$	$2e-7$	$4e-2$	7	$4e-09$	$9e-20$	$1e-1$
spikes	40.6	$1e-14$	$7e-4$	$1e+1$	200	$1e-06$	$1e-19$	$1e-5$
ursell	1.0	$5e-16$	$1e-7$	—	3	$4e-08$	$1e-16$	$5e-1$
wing	0.6	$6e-16$	$4e-7$	$6e-1$	3	$4e-11$	$7e-21$	$5e-4$

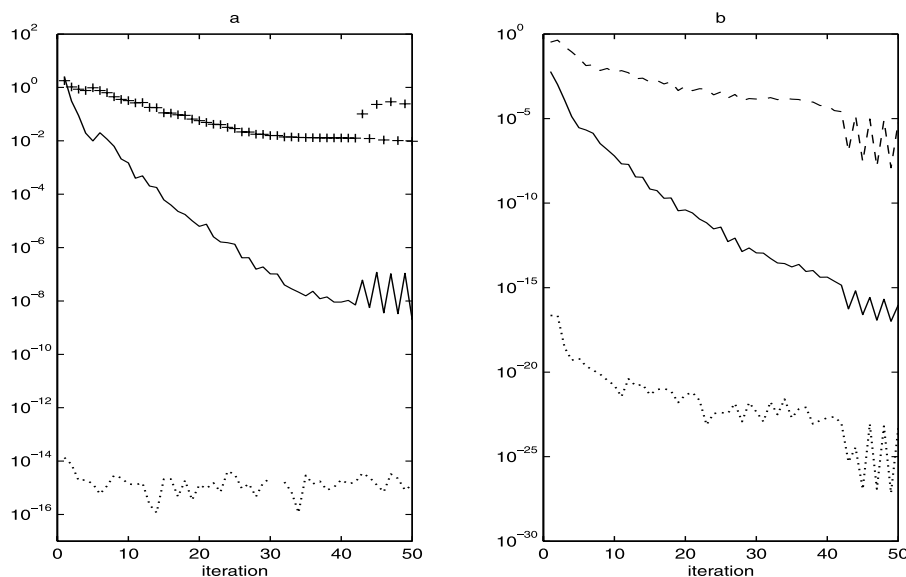


FIG. 7. Convergence of least squares solution for heat problem. Left:  $\|(H - \lambda)z_k - g\|/\|g\|$  (solid),  $(z_k^T z_k - \delta^2)/\delta^2$  (dot),  $\|z_k - x_{IP}\|/\|x_{IP}\|$  (+). Right:  $\|r_k\|$  (solid),  $|y_k^T r_k|$  (dot),  $|g^T y_k|$  (dash).

method due to Rojas and Sorenson [23]. The results also show improvement in accuracy in these tests. For the spike problem,  $\|r_k\|$  is of  $\mathcal{O}(1)$  throughout because of the large norm of  $H^2$  and hence  $\gamma_k, \eta_k$  are not reduced to the given thresholds.

We further observe from Table 2 that  $\gamma_k$  is proportional to  $\|r_k\|/|g^T y_k|$  and  $\eta_k$  is nearly proportional to  $|y_k^T r_k|/|g^T y_k|^2$ , as suggested by Theorem 4.1. With  $|g^T y_k|$  being very small in such problems, a typical iteration will see  $\|r_k\|$  gradually decreased, while  $|g^T y_k|$  is also decreased. Then,  $\gamma_k = \|(H - \theta I)z_k - g\|$  will stagnate at a level given by  $\delta^2 \|r_k\|/|g^T y_k|$ . On the other hand, with  $\theta_k$  computed through a Rayleigh quotient, a very good orthogonality  $|y_k^T r_k|$  is achieved and this in turn keeps  $\delta^2 |y_k^T r_k|/|g^T y_k|^2$  and hence the constraint error  $(z_k^T z_k - \delta^2)/\delta^2$  usually in the order of machine precision. So,  $z_k$  nearly satisfies the constraint throughout.

**6. Conclusions.** We have presented a basic Arnoldi-type process for a large monic quadratic matrix polynomial. The process is particularly efficient when some combination of the coefficient matrices  $A$  and  $B$  is of low rank, or one of them, say  $B$ , is a polynomial of  $A$  plus a low rank matrix. We have applied it to the quadratic eigenvalue problem arising in the quadratically constrained least squares problem. Our testing demonstrates its effectiveness for this class of problems.

**Acknowledgments.** The authors would like to acknowledge many fruitful conversations they had with Prof. Zhaojun Bai of the University of California at Davis during the course of this work. They are also indebted to the referees for their constructive and detailed suggestions that improved the paper significantly both in terms of presentations and technical details. In particular, they thank an anonymous referee for suggesting the Rayleigh quotient approach (4.8) to enforce the orthogonality between  $y$  and  $r$  from using the QR algorithm [11]. The approach also actually improves the solutions by the new process here. They also thank the other referee for suggesting the ordinary Lanczos process for (4.4) (see (4.9)).

## REFERENCES

- [1] W. E. ARNOLDI, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.
- [2] Z. BAI, *personal communication*, 2000.
- [3] Z. BAI, J. DEMMEL, J. DONGARRA, A. RUHE, AND H. VAN DER VORST, EDS., *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, 2000.
- [4] D. CALVETTI, G. H. GOLUB, AND L. REICHEL, *Estimation of the  $L$ -curve via Lanczos bidiagonalization*, BIT, 39 (1999), pp. 603–619.
- [5] D. CALVETTI, B. LEWIS, AND L. REICHEL, *On the regularizing properties of the GMRES method*, Numer. Math., 91 (2002), pp. 605–625.
- [6] D. CALVETTI, L. REICHEL, AND Q. ZHANG, *Iterative exponential filtering for large discrete ill-posed problems*, Numer. Math., 83 (1999), pp. 535–556.
- [7] J. W. DEMMEL, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
- [8] T. ERICSSON AND A. RUHE, *The spectral transformation Lanczos method for the numerical solution of large sparse generalized symmetric eigenvalue problems*, Math. Comp., 35 (1980), pp. 1251–1268.
- [9] G. E. FORSYTHE AND G. H. GOLUB, *On the stationary values of a second-degree polynomial on the unit sphere*, SIAM J. Soc. Indust. Appl. Math., 13 (1965), pp. 1050–1068.
- [10] W. GANDER, *Least squares with a quadratic constraint*, Numer. Math, 36 (1981) pp. 291–307.
- [11] W. GANDER, G. H. GOLUB, AND U. VON MATT, *A constrained eigenvalue problem*, Linear Algebra Appl., 114/115 (1989), pp. 815–839.
- [12] G. H. GOLUB AND U. VON MATT, *Quadratically constrained least squares and quadratic problems*, Numer. Math., 59 (1991). pp. 561–580.
- [13] I. GOHBERG, P. LANCASTER, AND L. RODMAN, *Matrix Polynomials*, Academic Press, New York, 1982.
- [14] G. H. GOLUB AND C. F. V. LOAN, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
- [15] P. C. HANSEN, *Regularization Tools: A MATLAB package for analysis and solution of discrete ill-posed problems*, Numer. Algorithms, 6 (1994), pp. 1–35.
- [16] P. C. HANSEN, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*, SIAM, Philadelphia, 1997.
- [17] L. HOFFNUNG, R.-C. LI, AND Q. YE, *Krylov Type Subspace Methods for Matrix Polynomials*, Research report 2002-08, Department of Mathematics, University of Kentucky, 2002, available online from <http://www.ms.uky.edu/~math/MAreport>; Linear Algebra Appl., to appear.
- [18] C. LANCZOS, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Research National Bureau Standards, 45 (1950), pp. 255–282.
- [19] R. LEHOUCQ, D. C. SORENSEN, AND C. YANG, *ARPACK User's Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, 1998.
- [20] K. MEERBERGEN, *Locking and restarting quadratic eigenvalue solvers*, SIAM J. Sci. Comput.,

- 22 (2001), pp. 1814–1839.
- [21] J. J. MORÉ AND D. C. SORESENSEN, *Computing a trust region step*, SIAM. J. Sci. Statist. Comput., 4 (1983), pp. 553–572.
  - [22] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, 1997.
  - [23] M. ROJAS AND D. C. SORESENSEN, *A trust-region approach to the regularization of large-scale discrete forms of ill-posed problems*, SIAM J. Sci. Comput., 23 (2002), pp. 1842–1860.
  - [24] M. ROJAS, S. A. SANTOS, AND D. C. SORESENSEN, *A new matrix-free algorithm for the large-scale trust-region subproblem*, SIAM J. Optim., 11 (2000), pp. 611–646
  - [25] G. SLEIJPEN, J. BOOTEN, D. FOKKEMA, AND H. VAN DER VORST, *Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems*, BIT, 36 (1996), pp. 593–633.
  - [26] D. C. SORESENSEN, *Newton’s method with a model trust region modification*, SIAM J. Numer. Anal., 19 (1982), pp. 409–426.
  - [27] D. C. SORESENSEN, *Minimization of a large-scale quadratic function subject to a spherical constraint*, SIAM J. Optim., 7 (1997). pp. 141–161.
  - [28] Y. SAAD, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, UK, 1992.
  - [29] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, MA, 1996.
  - [30] F. TISSEUR AND K. MEERBERGEN, *The quadratic eigenvalue problem*, SIAM Rev., 43 (2001), pp. 235–386.
  - [31] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, England, 1965.
  - [32] Q. YE, *A convergence analysis for nonsymmetric Lanczos algorithms*, Math. Comp., 56 (1991), pp. 677–691.
  - [33] T. ZHANG, G. H. GOLUB, AND K. H. LAW, *Eigenvalue perturbation and generalized Krylov subspace method*, Appl. Numer. Math., 27 (1998), pp. 185–202.