RESTARTED BLOCK LANCZOS BIDIAGONALIZATION METHODS

JAMES BAGLAMA* AND LOTHAR REICHEL†

Abstract. The problem of computing a few of the largest or smallest singular values and associated singular vectors of a large matrix arises in many applications. This paper describes restarted block Lanczos bidiagonalization methods based on augmentation of Ritz vectors or harmonic Ritz vectors by block Krylov subspaces.

Key words. partial singular value decomposition, restarted iterative method, implicit shifts, augmentation.

AMS subject classifications. 65F15, 15A18

1. Introduction.

Many problems in Applied Mathematics and Engineering require the computation of a few of the largest or smallest singular values and associated left and right singular vectors of a large matrix $A \in \mathbb{R}^{\ell \times n}$. These tasks arise, for instance, when one is interested in determining a low-rank approximation of A or when one wishes to compute the null space of A or of its transpose A^T. We may assume that $\ell \geq n$, because otherwise we replace A by A^T. Let

$$\sigma_1^{(A)} \geq \sigma_2^{(A)} \geq \ldots \geq \sigma_n^{(A)} \geq 0$$

denote the singular values of A, and let $u_j^{(A)} \in \mathbb{R}^\ell$ and $v_j^{(A)} \in \mathbb{R}^n$, $1 \leq j \leq n$, be associated left and right singular vectors, respectively. Hence,

$$Av_j^{(A)} = \sigma_j^{(A)} u_j^{(A)}, \qquad A^T u_j^{(A)} = \sigma_j^{(A)} v_j^{(A)}, \qquad 1 \leq j \leq n, \tag{1.1}$$

and

$$A = \sum_{j=1}^{n} \sigma_j^{(A)} u_j^{(A)} (v_j^{(A)})^T.$$

The matrices $U_n^{(A)} = [u_1^{(A)}, u_2^{(A)}, \ldots, u_n^{(A)}]$ and $V_n^{(A)} = [v_1^{(A)}, v_2^{(A)}, \ldots, v_n^{(A)}]$ have orthonormal columns. We refer to $\{\sigma_j^{(A)}, u_j^{(A)}, v_j^{(A)}\}$ as a singular triplet of A. Singular triplets associated with large (small) singular values are referred to as large (small) singular triplets.

This paper presents new restarted partial block Lanczos bidiagonalization methods for the computation of a few of the largest or smallest singular triplets. The methods determine sequences of partial block Lanczos bidiagonalizations of A associated with judiciously chosen initial blocks.

Application of m steps of block Lanczos bidiagonalization with block-size r to the matrix A with the initial block $P_r \in \mathbb{R}^{n \times r}$ with orthonormal columns yields the decompositions

$$AP_{mr} = Q_{mr} B_{mr}, \tag{1.2}$$

$$A^T Q_{mr} = P_{mr} B_{mr}^T + F_r E_r^T, \tag{1.3}$$

*Department of Mathematics, University of Rhode Island, Kingston, RI 02881. E-mail: jbaglama@math.uri.edu. Home page: http://www.math.uri.edu/∼jbaglama. Research supported in part by NSF grant DMS-0311786.

†Department of Mathematical Sciences, Kent State University, Kent, OH 44242. E-mail: reichel@math.kent.edu. Home page: http://www.math.kent.edu/∼reichel. Research supported in part by NSF grant DMS-0107858 and an OBR Research Challenge Grant.

where $P_{mr} \in \mathbb{R}^{n\times mr}$, $Q_{mr} \in \mathbb{R}^{\ell\times mr}$, $P_{mr}^T P_{mr} = Q_{mr}^T Q_{mr} = I_{mr}$, and I_{mr} denotes the identity matrix of order mr. The matrix P_{mr} has the leading $n \times r$ submatrix P_r. We refer to $F_r \in \mathbb{R}^{n\times r}$ as the residual matrix. It satisfies

$$P_{mr}^T F_r = 0. \tag{1.4}$$

The matrix

$$B_{mr} := \begin{bmatrix} S^{(1)} & L^{(2)} & & & & 0 \\ & S^{(2)} & L^{(3)} & & & \\ & & S^{(3)} & & & \\ & & & \ddots & & \\ & & & & \ddots & L^{(m)} \\ 0 & & & & & S^{(m)} \end{bmatrix} \in \mathbb{R}^{mr\times mr} \tag{1.5}$$

is upper block-bidiagonal with upper triangular diagonal blocks $S^{(j)} \in \mathbb{R}^{r\times r}$ and lower triangular superdiagonal blocks $L^{(j)} \in \mathbb{R}^{r\times r}$. Thus, B_{mr} is upper triangular. Throughout this paper, the matrix E_r consists of the r last columns of an identity matrix of appropriate order. We refer to the decompositions (1.2)-(1.3) as a partial block Lanczos bidiagonalization of A. The number of block-bidiagonalization steps, m, is assumed to be small enough so that the decompositions (1.2)-(1.3) with the stated properties exist. The matrix (1.5) is assumed to be sufficiently small to allow easy computation of its singular triplets $\{\sigma_j^{(B_{mr})}, u_j^{(B_{mr})}, v_j^{(B_{mr})}\}_{j=1}^{mr}$. Note that when the residual matrix F_r vanishes, the singular values of B_{mr} are singular values of A, and the associated singular vectors of A can be determined from P_{mr}, Q_{mr}, and the singular vectors of B_{mr}. Moreover, when F_r is of small norm, the singular values $\{\sigma_j^{(B_{mr})}\}_{j=1}^{mr}$ are close to singular values of A; see Section 2 for further details.

For large matrices A, i.e., when ℓ, and possibly also n, are large, the storage requirement of the partial block Lanczos bidiagonalization (1.2)-(1.3) is large, unless mr is small. However, when the number of block Lanczos steps m and the block-size r are chosen so that mr is small, the singular values of B_{mr} may not furnish sufficiently accurate approximations of the desired singular values of A. This difficulty can be circumvented by letting mr be fairly small and computing a sequence of partial block Lanczos bidiagonalizations (1.2)-(1.3) associated with judiciously chosen initial blocks P_r. Methods based on this approach are known as restarted block Lanczos bidiagonalization methods and will be discussed in the present paper.

The first restarted block Lanczos bidiagonalization method for computing a few of the largest singular triplets of a large matrix is described by Golub et al. [6]. During the last few years considerable progress has been made in the development of restarted methods, based on the Lanczos and Arnoldi processes, for the computation of a few eigenvalues and associated eigenvectors of a large sparse matrix; see, e.g., Lehoucq [12] and Sorensen [20] for recent discussions of this type of methods. The latter methods have spurred the development of new restarted Lanczos bidiagonalization methods with block-size one; see, e.g., Björck et al. [5], Jia and Niu [10], Kokiopoulou et al. [11], as well as [2].

The present paper extends the restarted Lanczos bidiagonalization methods in [2] to block-methods. Our interest in block-methods stems from the fact that they can detect and compute close singular values and associated singular vectors more reliably than the corresponding methods with block-size one. This property of block-methods has been discussed by Parlett [18] in the context of eigenvalue computation of large symmetric matrices. Numerical illustrations for singular value computation are presented in Section 5.

We also remark that for some large problems, the computer time required to evaluate products of the matrices A or A^T with blocks of r vectors, while large, only grows quite slowly with r. This situation arises, for instance, when the matrix A is not explicitly stored and its entries are computed when required for the evaluation of matrix-vector products with A and A^T. Block-methods then are the methods of choice. Moreover, block-size $r > 1$ makes it possible to describe much of the arithmetic work in terms of sparse BLAS, which on many computers can be executed quite efficiently; see Vudoc et al. [21] and Wijshoff [22].

This paper is organized as follows. Section 2 describes an algorithm for the computation of the partial block Lanczos bidiagonalization (1.2)-(1.3) and introduces notation used throughout the paper. A restarted block Lanczos bidiagonalization method based on augmentation of Ritz vectors by a block Krylov subspace is presented in Section 3, and Section 4 describes an analogous scheme based on augmentation of harmonic Ritz vectors. These methods are block-generalizations of schemes discussed in [2]. Computed examples can be found in Section 5 and concluding remarks in Section 6.

When a good preconditioner is available for the solution of linear systems of equations with the matrix A, methods of Jacobi-Davidson-type are attractive for the computation of a few of the largest or smallest singular triplets; see Hochstenbach [8, 9]. The methods of the present paper are suited for matrices for which no good preconditioner is known.

2. Partial block Lanczos bidiagonalization. The following algorithm determines the partial block Lanczos bidiagonalization (1.2)-(1.3) of $A \in \mathbb{R}^{\ell\times n}$. The number of block Lanczos bidiagonalization steps m and the block-size r typically are chosen so that $mr \ll \min\{\ell, n\}$. The algorithm as stated requires that all the triangular $r\times r$ matrices $L^{(j+1)}$ and $S^{(j+1)}$ generated be nonsingular, however, the MATLAB code used for the computed examples does not impose this restriction; see the discussion below. The jth step of the algorithm determines sets of r new columns of the matrices P_{mr} and Q_{mr}. We refer to these sets as $P^{(j+1)} \in \mathbb{R}^{n\times r}$ and $Q^{(j+1)} \in \mathbb{R}^{\ell\times r}$, respectively. Thus,

$$P_{mr} = [P^{(1)}, P^{(2)}, \ldots, P^{(m)}], \qquad Q_{mr} = [Q^{(1)}, Q^{(2)}, \ldots, Q^{(m)}].$$

Throughout this paper $\|\cdot\|$ denotes the Euclidean vector norm or the associated induced matrix norm.

Algorithm 2.1. Block Lanczos Bidiagonalization

Input: *matrix* $A \in \mathbb{R}^{\ell\times n}$ *or functions for evaluating matrix-vector products with* A *and* A^T*,*
r : *block-size* ≥ 1*,*
$P_r \in \mathbb{R}^{n\times r}$: *initial block with* r *orthonormal columns,*
m : *number of bidiagonalization steps.*

Output: $P_{mr} = [P^{(1)}, P^{(2)}, \ldots, P^{(m)}] \in \mathbb{R}^{n\times mr}$: *matrix with orthonormal columns,*
$Q_{mr} = [Q^{(1)}, Q^{(2)}, \ldots, Q^{(m)}] \in \mathbb{R}^{\ell\times mr}$: *matrix with orthonormal columns,*

$B_{mr} \in \mathbb{R}^{mr \times mr}$: upper block-bidiagonal matrix (1.5),
$F_r \in \mathbb{R}^{n \times r}$: residual matrix.

1. *$P^{(1)} := P_r$; $W_r := AP^{(1)}$;*
2. *Compute QR-factorization: $W_r =: Q^{(1)} S^{(1)}$ ($S^{(1)}$ is upper triangular);*
3. *for $j = 1 : m$*
    4. *$F_r := A^T Q^{(j)} - P^{(j)} (S^{(j)})^T$;*
    5. *Reorthogonalization: $F_r := F_r - P_{jr}(P_{jr}^T F_r)$;*
    6. *if $j < m$ then*
        7. *Compute QR-factorization: $F_r =: P^{(j+1)} R^{(j+1)}$; $L^{(j+1)} := (R^{(j+1)})^T$;*
        8. *$W_r := AP^{(j+1)} - Q^{(j)} L^{(j+1)}$;*
        9. *Reorthogonalization: $W_r := W_r - Q_{jr}(Q_{jr}^T W_r)$;*
        10. *Compute QR-factorization: $W_r =: Q^{(j+1)} S^{(j+1)}$;*
    11. *endif*
12. *endfor*

The algorithm is formulated for matrices A with real-valued entries; however, the MATLAB code used for the numerical examples reported in Section 5 can be applied to matrices with complex-valued entries as well. In the latter case, transposition is followed by complex conjugation.

When the computations with Algorithm 2.1 are carried out in finite precision arithmetic and the matrices F_r and W_r are not reorthogonalized against the columns of the available matrices $P_{jr} = [P^{(1)}, P^{(2)}, \ldots, P^{(j)}]$ and $Q_{jr} = [Q^{(1)}, Q^{(2)}, \ldots, Q^{(j)}]$, respectively, the computed columns of P_{mr} and Q_{mr} might be far from orthogonal. We therefore reorthogonalize in lines 5 and 9 of the algorithm.

Several reorthogonalization strategies for the columns of the matrices P_{mr} and Q_{mr} are discussed in the literature for the case when the block-size r is one; see [2] for a recent review and references. Here we only note that Simon and Zha [19] observed that when the matrix A is not very ill-conditioned, only the columns of one of the matrices P_{mr} or Q_{mr} need to be reorthogonalized. Reorthogonalization of the columns of P_{mr} only can reduce the computational effort required to compute the decompositions (1.2)-(1.3) considerably when $\ell \gg n$. Algorithm 2.1 easily can be modified to reorthogonalize only the columns of P_{mr}. Our MATLAB code allows a user to choose between reorthogonalization of the columns of both P_{mr} and Q_{mr}, and reorthogonalization of the columns of P_{mr} only.

The QR-factorizations in Algorithm 2.1 and elsewhere in our MATLAB code are computed by the MATLAB function `qr` with column pivoting. For instance, in step 2 of the algorithm, the function `qr` yields the factorization $W_r = Q^{(1)} S^{(1)}$, where $Q^{(1)} \in \mathbb{R}^{\ell \times r}$ has orthonormal columns and $S^{(1)} \in \mathbb{R}^{r \times r}$ is upper triangular up to a column permutation. Such a factorization is computed also when the columns of W_r are linearly dependent or nearly so. For simplicity, we will refer to the matrices $S^{(j+1)}$ and $L^{(j+1)}$, $j = 0, 1, \ldots, m-1$, determined by Algorithm 2.1 as upper and lower triangular, respectively, even if, due to possible column permutation, they might not be. Thus, Algorithm 2.1 computes the desired output even for problems that give rise to matrices W_r of less than full rank.

Let $S^{(1)} = [s_{jk}]_{j,k=1}^{r}$ be the triangular matrix obtained by QR-factorization of W_r with column pivoting, and assume for notational simplicity that $S^{(1)}$ really is upper triangular. Then the diagonal entries satisfy $|s_{11}| \geq |s_{22}| \geq \ldots \geq |s_{rr}|$, and they can, for small block-sizes r, be used to compute fairly accurate bounds for the condition number $\kappa(W_r)$ of W_r. Here $\kappa(W_r) := \|W_r\| \, \|W_r^\dagger\|$, where $W_r^\dagger$ denotes the Moore-Penrose pseudoinverse of W_r. It follows from Björck [4, Section 2.7.3] that

$$\frac{|s_{11}|}{|s_{rr}|} \leq \kappa(W_r) \leq 2^{r-1}\sqrt{r}\frac{|s_{11}|}{|s_{rr}|}. \tag{2.1}$$

We repeat steps 9 and 10 of Algorithm 2.1 when the left-hand side of (2.1) is large in order to secure that the columns of $Q^{(j+1)}$, that may have been introduced by the MATLAB function `qr` and are not in the range of W_r, are numerically orthogonal to the range of Q_{jr}. We remark that the occurrence of a large left-hand side in (2.1) is rare.

It follows from Algorithm 2.1 that $L^{(m+1)}$, the last superdiagonal block of the upper block-bidiagonal matrix $B_{(m+1)r} \in \mathbb{R}^{(m+1)r \times (m+1)r}$, which is obtained by $m+1$ block Lanczos bidiagonalization steps, can be computed by QR-factorization of the residual matrix F_r determined by the algorithm in step m, i.e.,

$$F_r =: P^{(m+1)} R^{(m+1)}, \qquad L^{(m+1)} := (R^{(m+1)})^T, \tag{2.2}$$

where $P^{(m+1)} \in \mathbb{R}^{n \times r}$ has orthonormal columns and $R^{(m+1)} \in \mathbb{R}^{r \times r}$ is upper triangular. Thus, the matrix $P^{(m+1)}$, which makes up the last r columns of $P_{(m+1)r}$, can be computed when the decompositions (1.2)-(1.3) are available. The relation (1.3) can be expressed as

$$A^T Q_{mr} = P_{(m+1)r} B_{mr,(m+1)r}^T, \tag{2.3}$$

where the matrix $B_{mr,(m+1)r} \in \mathbb{R}^{mr \times (m+1)r}$ is obtained by appending the columns $E_r L^{(m+1)}$ to B_{mr}. Note that $B_{mr,(m+1)r}$ is the leading $mr \times (m+1)r$ submatrix of the matrix $B_{(m+1)r}$, which is obtained after $m+1$ steps of block Lanczos bidiagonalization.

We remark that a right singular vector of A associated with a zero singular value can be expressed as a linear combination of the columns of P_{mr}, e.g., by using the singular value decomposition of B_{mr}. However, the corresponding left singular vector of A is not readily available.

We will use the connection between partial block Lanczos bidiagonalization (1.2)-(1.3) of A and partial block Lanczos tridiagonalization of the matrix $A^T A$. Multiplying equation (1.2) by A^T from the left-hand side yields

$$A^T A P_{mr} = P_{mr} B_{mr}^T B_{mr} + F_r E_r^T B_{mr}. \tag{2.4}$$

The matrix

$$T_{mr} := B_{mr}^T B_{mr} \in \mathbb{R}^{mr \times mr} \tag{2.5}$$

is symmetric and block-tridiagonal with block-size r, and the expression (2.4) is a partial block Lanczos tridiagonalization of $A^T A$ with initial block-vector P_r; see, e.g., [7, Section 9.2.6] for a discussion on block Lanczos tridiagonalization. Since T_{mr} is block-tridiagonal, equation (2.4) shows that the block-columns $P^{(j)}$ of P_{mr} satisfy a three-term recurrence relation. Moreover, the block-columns $P^{(1)}, P^{(2)}, \ldots, P^{(m)}$ form an orthonormal basis of the block Krylov subspace

$$\mathbb{K}_m(A^T A, P_r) = \text{span}\{P_r, A^T A P_r, (A^T A)^2 P_r, \ldots, (A^T A)^{m-1} P_r\}. \tag{2.6}$$

Similarly, multiplying (1.3) by A from the left-hand side yields

$$A A^T Q_{mr} = Q_{mr} B_{mr} B_{mr}^T + A F_r E_r^T. \tag{2.7}$$

The columns of Q_{mr} form an orthonormal basis of the block Krylov subspace

$$\mathbb{K}_m(AA^T, Q_r) = \text{span}\{Q_r, AA^TQ_r, (AA^T)^2Q_r, \dots, (AA^T)^{m-1}Q_r\} \tag{2.8}$$

with $Q_r := AP_r$. We remark that since the columns of Q_{mr} generally are not orthogonal to AF_r, the decomposition (2.7) typically is not a block Lanczos tridiagonalization of AA^T.

Let $\{\sigma_j^{(B_{mr})}, u_j^{(B_{mr})}, v_j^{(B_{mr})}\}_{j=1}^{mr}$ denote the singular triplets of B_{mr} enumerated so that

$$\sigma_1^{(B_{mr})} \geq \sigma_2^{(B_{mr})} \geq \ldots \geq \sigma_{mr}^{(B_{mr})} \geq 0. \tag{2.9}$$

Then, analogously to (1.1),

$$B_{mr}v_j^{(B_{mr})} = \sigma_j^{(B_{mr})}u_j^{(B_{mr})}, \quad B_{mr}^T u_j^{(B_{mr})} = \sigma_j^{(B_{mr})}v_j^{(B_{mr})}, \quad 1 \leq j \leq mr, \tag{2.10}$$

and the $mr \times mr$ matrices of left and right singular vectors

$$U_{mr}^{(B_{mr})} := [u_1^{(B_{mr})}, u_2^{(B_{mr})}, \dots, u_{mr}^{(B_{mr})}], \quad V_{mr}^{(B_{mr})} := [v_1^{(B_{mr})}, v_2^{(B_{mr})}, \dots, v_{mr}^{(B_{mr})}],$$

are orthogonal.

We determine approximate singular triplets $\{\tilde{\sigma}_j^{(A)}, \tilde{u}_j^{(A)}, \tilde{v}_j^{(A)}\}_{j=1}^{mr}$ of A from the singular triplets of B_{mr} by

$$\tilde{\sigma}_j^{(A)} := \sigma_j^{(B_{mr})}, \qquad \tilde{u}_j^{(A)} := Q_{mr}u_j^{(B_{mr})}, \qquad \tilde{v}_j^{(A)} := P_{mr}v_j^{(B_{mr})}. \tag{2.11}$$

Combining (2.10) and (2.11) with (1.2)-(1.3) shows that

$$A\tilde{v}_j^{(A)} = \tilde{\sigma}_j^{(A)}\tilde{u}_j^{(A)}, \quad A^T\tilde{u}_j^{(A)} = \tilde{\sigma}_j^{(A)}\tilde{v}_j^{(A)} + F_rE_r^Tu_j^{(B_{mr})}, \quad 1 \leq j \leq mr. \tag{2.12}$$

It follows from the orthonormality of the columns of the matrices P_{mr} and $U_{mr}^{(B_{mr})}$ that the approximate left singular vectors $\tilde{u}_1^{(A)}, \tilde{u}_2^{(A)}, \dots, \tilde{u}_{mr}^{(A)}$ are orthonormal. Similarly, the approximate right singular vectors $\tilde{v}_1^{(A)}, \tilde{v}_2^{(A)}, \dots, \tilde{v}_{mr}^{(A)}$ are also orthonormal.

The relations (2.12) suggest that an approximate singular triplet $\{\tilde{\sigma}_j^{(A)}, \tilde{u}_j^{(A)}, \tilde{v}_j^{(A)}\}$ be accepted as a singular triplet of A if $F_rE_r^Tu_j^{(B_{mr})}$ is sufficiently small. Specifically, our numerical method accepts $\{\tilde{\sigma}_j^{(A)}, \tilde{u}_j^{(A)}, \tilde{v}_j^{(A)}\}$ as a singular triplet of A if

$$\|R^{(m+1)}\| \, \|E_r^Tu_j^{(B_{mr})}\| \leq \delta\|A\| \tag{2.13}$$

for a user-specified value of δ, where we have used (2.2). The quantity $\|A\|$ in (2.13) is easily approximated by the largest singular value $\sigma_1^{(B_{mr})}$ of the block-bidiagonal matrix B_{mr}. The computation of $\sigma_1^{(B_{mr})}$ is inexpensive because the matrix B_{mr} is not large. During the computations of the desired singular triplets of A, typically, several matrices B_{mr} and their singular value decompositions are computed. We approximate $\|A\|$ by the largest of the singular values of all the matrices B_{mr} generated. This generally gives a good estimate of $\|A\|$.

3. Restarting by augmentation of Ritz vectors. Wu and Simon [23] recently discussed the computation of a few extreme eigenvalues of a large symmetric matrix by a restarted Lanczos tridiagonalization method, and proposed an implementation based on the augmentation of Ritz vectors by a Krylov subspace. A restarted Lanczos bidiagonalization method based on this approach is described in [2]. The present section generalizes the latter scheme to allow block-size r larger than one.

Let the partial block Lanczos bidiagonalization (1.2)-(1.3) be available, and assume that we are interested in determining the k largest singular triplets of A, where $k < mr$. Consider the approximate right singular vectors $\tilde{v}_j^{(A)}$, $1 \le j \le mr$, determined by (2.11). It follows from (2.2), (2.4), (2.10), and (2.11), that

$$A^T A \tilde{v}_j^{(A)} - (\tilde{\sigma}_j^{(A)})^2 \tilde{v}_j^{(A)} = \tilde{\sigma}_j^{(A)} P^{(m+1)} R^{(m+1)} E_r^T u_j^{(B_{mr})}, \qquad 1 \le j \le mr, \tag{3.1}$$

which shows that the residual errors for all vectors $\tilde{v}_j^{(A)}$, $1 \le j \le mr$, live in $\mathcal{R}(P^{(m+1)})$, the range of $P^{(m+1)}$. Moreover, since the vectors $\tilde{v}_j^{(A)}$ are orthogonal to $\mathcal{R}(P^{(m+1)})$, they are Ritz vectors of $A^T A$. Specifically, $\tilde{v}_j^{(A)}$ is a Ritz vector associated with the Ritz value $(\tilde{\sigma}_j^{(A)})^2$. If $\tilde{\sigma}_j^{(A)} = 0$, then $\tilde{v}_j^{(A)}$ is an (exact) eigenvector of $A^T A$.

We now derive modifications of the decompositions (1.2)-(1.3), in which the k first columns of an analog of the matrix P_{mr} are the Ritz vectors $\tilde{v}_1^{(A)}, \tilde{v}_2^{(A)}, \ldots, \tilde{v}_k^{(A)}$ associated with the k largest Ritz values. We recall that these Ritz vectors are also approximate right singular vectors of A. Their accuracy, as well as the accuracy of available approximate left singular vectors, is improved by augmenting the modified decompositions by block Krylov subspaces, and then restarting the computations.

Let the Ritz vectors $\tilde{v}_j^{(A)}$, $1 \le j \le k$, be available and introduce the matrix

$$\tilde{P}_{k+r} := [\tilde{v}_1^{(A)}, \tilde{v}_2^{(A)}, \ldots, \tilde{v}_k^{(A)}, P^{(m+1)}] \in \mathbb{R}^{n \times (k+r)}. \tag{3.2}$$

It follows from (2.11) that

$$A\tilde{P}_{k+r} = [\tilde{\sigma}_1^{(A)} \tilde{u}_1^{(A)}, \tilde{\sigma}_2^{(A)} \tilde{u}_2^{(A)}, \ldots, \tilde{\sigma}_k^{(A)} \tilde{u}_k^{(A)}, AP^{(m+1)}]. \tag{3.3}$$

Orthogonalization of $AP^{(m+1)}$ against the vectors $\tilde{u}_j^{(A)}$, $1 \le j \le k$, yields

$$AP^{(m+1)} = \sum_{j=1}^{k} \tilde{u}_j^{(A)} \tilde{r}_j^T + W^{(m+1)}, \tag{3.4}$$

where $\tilde{r}_j^T := (\tilde{u}_j^{(A)})^T A P^{(m+1)}$, $1 \le j \le k$, and the columns of the remainder matrix $W^{(m+1)} \in \mathbb{R}^{\ell \times r}$ are orthogonal to the vectors $\tilde{u}_j^{(A)}$, $1 \le j \le k$. The vectors $\tilde{r}_j$ can be evaluated inexpensively by using the right-hand side of

$$\tilde{r}_j = (P^{(m+1)})^T A^T \tilde{u}_j^{(A)} = (P^{(m+1)})^T (\tilde{\sigma}_j^{(A)} \tilde{v}_j^{(A)} + F_r E_r^T u_j^{(B_{mr})}) = R^{(m+1)} E_r^T u_j^{(B_{mr})}.$$

Introduce the QR-factorization

$$W^{(m+1)} =: Q^{(m+1)} S^{(m+1)},$$

where $Q^{(m+1)} \in \mathbb{R}^{\ell \times r}$ has orthonormal columns and $S^{(m+1)} \in \mathbb{R}^{r \times r}$ is upper triangular. We remark that similarly as in Section 2, column pivoting is applied when evaluating the QR-factorization. Therefore $S^{(m+1)}$ is upper triangular up to column pivoting, only. This comment applies to all "triangular" factors of QR-factorizations computed throughout this paper.

Define the matrices

$$\tilde{Q}_{k+r} := [\tilde{u}_1^{(A)}, \tilde{u}_2^{(A)}, \ldots, \tilde{u}_k^{(A)}, Q^{(m+1)}] \in \mathbb{R}^{\ell\times(k+r)} \tag{3.5}$$

and

$$\tilde{B}_{k+r} := \begin{bmatrix} \tilde{\sigma}_1^{(A)} & & 0 & \tilde{r}_1^T \\ & \ddots & & \vdots \\ & & \tilde{\sigma}_k^{(A)} & \tilde{r}_k^T \\ 0 & & & S^{(m+1)} \end{bmatrix} \in \mathbb{R}^{(k+r)\times(k+r)}. \tag{3.6}$$

The matrix $\tilde{Q}_{k+r}$ has orthonormal columns and $\tilde{B}_{k+r}$ may have nonvanishing entries only on the diagonal and in the last r columns. Substituting (3.4) into (3.3) yields the decomposition

$$A\tilde{P}_{k+r} = \tilde{Q}_{k+r}\tilde{B}_{k+r}, \tag{3.7}$$

which is our desired analog of (1.2).

We turn to the matrix

$$A^T\tilde{Q}_{k+r} = [A^T\tilde{u}_1^{(A)}, A^T\tilde{u}_2^{(A)}, \ldots, A^T\tilde{u}_k^{(A)}, A^TQ^{(m+1)}], \tag{3.8}$$

which we would like to express in terms of $\tilde{P}_{k+r}$ and $\tilde{B}_{k+r}^T$. This will give an analogue of the decomposition (1.3). The first k columns of (3.8) are linear combinations of the vectors $\tilde{v}_j^{(A)}$ and the columns of $P^{(m+1)}$; specifically, we have for $1 \le j \le k$,

$$A^T\tilde{u}_j^{(A)} = \tilde{\sigma}_j^{(A)}\tilde{v}_j^{(A)} + P^{(m+1)}R^{(m+1)}E_r^Tu_j^{(B_{mr})} = \tilde{\sigma}_j^{(A)}\tilde{v}_j^{(A)} + P^{(m+1)}\tilde{r}_j. \tag{3.9}$$

The last r columns of (3.8) are orthogonal to the Ritz vectors $\tilde{v}_j^{(A)}$,

$$(\tilde{v}_j^{(A)})^TA^TQ^{(m+1)} = \tilde{\sigma}_j^{(A)}(\tilde{u}_j^{(A)})^TQ^{(m+1)} = 0, \qquad 1 \le j \le k.$$

Therefore they can be expressed as

$$A^TQ^{(m+1)} = P^{(m+1)}Z^{(m+1)} + \tilde{F}_r, \qquad Z^{(m+1)} \in \mathbb{R}^{r\times r}, \tag{3.10}$$

where the columns of $\tilde{F}_r \in \mathbb{R}^{n\times r}$ are orthogonal to the vectors $\tilde{v}_j^{(A)}$, $1 \le j \le k$, and to $\mathcal{R}(P^{(m+1)})$. Since

$$(Q^{(m+1)})^TAP^{(m+1)} = (Q^{(m+1)})^T(\sum_{j=1}^k \tilde{u}_j^{(A)}\tilde{r}_j^T + W^{(m+1)}) = S^{(m+1)},$$

it follows from (3.10) that $Z^{(m+1)} = (S^{(m+1)})^T$. This observation, together with (3.9) and (3.10), shows that

$$A^T\tilde{Q}_{k+r} = \tilde{P}_{k+r}\tilde{B}_{k+r}^T + \tilde{F}_rE_r^T, \tag{3.11}$$

which is the desired analogue of the decomposition (1.3). We remark that $\tilde{F}_r$ can be computed from (3.10).

If the matrix $\tilde{F}_r$ in (3.11) vanishes, then the singular values of $\tilde{B}_{k+r}$ are singular values of A, and we are done. When $\tilde{F}_r \neq 0$, new block-columns $P^{(m+j)} \in \mathbb{R}^{n\times r}$ and $Q^{(m+j)} \in \mathbb{R}^{\ell\times r}$, $j = 2, 3, \dots, \ell$, are computed and appended to the matrices $\tilde{P}_{k+r}$ and $\tilde{Q}_{k+r}$, respectively, as follows. Evaluate the QR-factorization

$$\tilde{F}_r =: P^{(m+2)} R^{(m+2)}, \tag{3.12}$$

where $P^{(m+2)} \in \mathbb{R}^{n\times r}$ has orthonormal columns and $R^{(m+2)} \in \mathbb{R}^{r\times r}$ is upper triangular. Let $L^{(m+2)} := (R^{(m+2)})^T$. Note that the matrix $\tilde{P}_{k+2r} := [\tilde{P}_{k+r}, P^{(m+2)}] \in \mathbb{R}^{n\times(k+2r)}$ has orthonormal columns. Determine the QR-factorization

$$Q^{(m+2)} S^{(m+2)} := (I - \tilde{Q}_{k+r}\tilde{Q}_{k+r}^T) A P^{(m+2)}, \tag{3.13}$$

where $Q^{(m+2)} \in \mathbb{R}^{\ell\times r}$ has orthonormal columns and $S^{(m+2)} \in \mathbb{R}^{r\times r}$ is upper triangular. Substituting the transpose of equation (3.11) into the right-hand side of (3.13), and using (3.12), shows that

$$Q^{(m+2)} S^{(m+2)} = A P^{(m+2)} - Q^{(m+1)} L^{(m+2)}. \tag{3.14}$$

Let $\tilde{Q}_{k+2r} := [\tilde{Q}_{k+r}, Q^{(m+2)}] \in \mathbb{R}^{\ell\times(k+2r)}$ and define the matrix $\tilde{B}_{k+2r}$ by first appending the column $E_r L^{(m+2)}$ and then the row $[0, \dots, 0, S^{(m+2)}] \in \mathbb{R}^{r\times(k+2r)}$ to $\tilde{B}_{k+r}$, i.e.,

$$\tilde{B}_{k+2r} := \begin{bmatrix} \tilde{\sigma}_1^{(A)} & & 0 & \tilde{r}_1^T & \\ & \ddots & & \vdots & 0 \\ & & \tilde{\sigma}_k^{(A)} & \tilde{r}_k^T & \\ & & & S^{(m+1)} & L^{(m+2)} \\ 0 & & & & S^{(m+2)} \end{bmatrix} \in \mathbb{R}^{(k+2r)\times(k+2r)}.$$

It now follows from (3.7) and (3.14) that

$$A\tilde{P}_{k+2r} = \tilde{Q}_{k+2r}\tilde{B}_{k+2r}. \tag{3.15}$$

We proceed by evaluating the QR-factorization

$$P^{(m+3)} R^{(m+3)} := (I - \tilde{P}_{k+2r}\tilde{P}_{k+2r}^T) A^T Q^{(m+2)}, \tag{3.16}$$

where $P^{(m+3)} \in \mathbb{R}^{n\times r}$ has orthonormal columns and $R^{(m+3)} \in \mathbb{R}^{r\times r}$ is upper triangular. Let $L^{(m+3)} := (R^{(m+3)})^T$. Substituting (3.15) into (3.16) gives

$$P^{(m+3)} R^{(m+3)} = A^T Q^{(m+2)} - P^{(m+2)} (S^{(m+2)})^T,$$

which shows that

$$A^T \tilde{Q}_{k+2r} = \tilde{P}_{k+2r}\tilde{B}_{k+2r}^T + P^{(m+3)} R^{(m+3)} E_r^T. \tag{3.17}$$

The decompositions (3.15) and (3.17) are analogous to (3.7) and (3.11). We continue in this manner to append new block-columns to the matrices $\tilde{P}_{k+jr}$ and $\tilde{Q}_{k+jr}$, as well as rows and columns to the matrices $\tilde{B}_{k+jr}$, for $j = 2, 3, \ldots, i-1$, to obtain the decompositions

$$A\tilde{P}_{k+ir} = \tilde{Q}_{k+ir}\tilde{B}_{k+ir}, \quad A^T\tilde{Q}_{k+ir} = \tilde{P}_{k+ir}\tilde{B}_{k+ir}^T + P^{(m+i+1)}R^{(m+i+1)}E_r^T, \tag{3.18}$$

where the matrices $\tilde{P}_{k+ir}$ and $\tilde{Q}_{k+ir}$ have orthonormal columns and

$$\tilde{B}_{k+ir} = \begin{bmatrix} \tilde{\sigma}_1^{(A)} & & 0 & \tilde{r}_1^T & & & \\ & \ddots & & \vdots & & 0 & \\ & & \tilde{\sigma}_k^{(A)} & \tilde{r}_k^T & & & \\ & & & S^{(m+1)} & L^{(m+2)} & & \\ & & & & S^{(m+2)} & \ddots & \\ & 0 & & & & \ddots & L^{(m+i)} \\ & & & & & & S^{(m+i)} \end{bmatrix} \in \mathbb{R}^{(k+ir)\times(k+ir)}.$$

Typically, the number of steps i is chosen so that the matrices in the decompositions (3.18) require about the same amount of computer storage as the matrices in the decompositions (1.2) and (1.3), i.e., $i \approx m - k/r$.

Having determined the decompositions (3.18), the method proceeds by computing the singular value decomposition of $\tilde{B}_{k+ir}$. The k largest singular triplets of $\tilde{B}_{k+ir}$ give us approximations of the k largest singular triplets of A by using formulas analogous to (2.11). These k largest approximate triplets determine new decompositions of the form (3.7) and (3.11), from which we compute new decompositions of the form (3.18). The computations continue in this manner until sufficiently accurate approximations of the k largest singular triplets of A have been found. An algorithm that describes these computations for block-size $r = 1$ can be found in [2, Subsection 3.3].

This section has discussed how to determine approximations of the k largest singular triplets of A. The computations of approximations of the k smallest singular triplets of A can be carried out analogously. The vectors $\tilde{v}_j^{(A)}$, $1 \leq j \leq k$, in (3.2) then should be right approximate singular vectors of the k smallest available approximate singular triplets of A (instead of the largest ones). We note, however, that when the k smallest singular triplets of A are desired, it may be advantageous to augment harmonic Ritz vectors instead of Ritz vectors by block Krylov subspaces. This approach is described in the following section.

4. Restarting by augmentation of harmonic Ritz vectors. Numerical examples reported in [2] for block-size $r = 1$ show that augmentation of harmonic Ritz vectors may yield faster convergence towards the smallest singular triplets than augmentation of Ritz vectors. Kokiopoulou et al. [11] report analogous behavior of restarted Lanczos bidiagonalization methods that apply implicit shifts. This section generalizes the method in [2] for augmentation of harmonic Ritz vectors to block-size r larger than one.

Let the partial Lanczos bidiagonalization (1.2)-(1.3) of A be available and assume that the matrix B_{mr} is nonsingular. The harmonic Ritz values $\hat{\theta}_j$ of A^TA associated with the partial block Lanczos tridiagonalization (2.4) are the eigenvalues of the generalized eigenvalue problem

$$((B_{mr}^TB_{mr})^2 + E_r(S^{(m)})^TL^{(m+1)}(L^{(m+1)})^TS^{(m)}E_r^T)\hat{w}_j = \hat{\theta}_jB_{mr}^TB_{mr}\hat{w}_j, \tag{4.1}$$

with $\hat{w}_j \in \mathbb{R}^{mr}\backslash\{0\}$ for $1 \leq j \leq mr$; see, e.g., Morgan [14, 15], Paige et al. [17], and [1] for properties of harmonic Ritz values.

The eigenpairs $\{\hat{\theta}_j, \hat{w}_j\}$ of (4.1) can be determined from the singular value decomposition of the matrix $B_{mr,(m+1)r}$ introduced in (2.3) as follows. Let $\{\sigma'_j, u'_j, v'_j\}_{j=1}^{mr}$ denote the singular triplets of $B_{mr,(m+1)r}$. We enumerate the triplets so that

$$0 < \sigma'_1 \leq \sigma'_2 \leq \ldots \leq \sigma'_{mr}, \tag{4.2}$$

because in the present section, we are concerned with the computation of the $k < mr$ smallest singular triplets of A. The k smallest singular triplets of $B_{mr,(m+1)r}$ determine the matrices

$$\begin{aligned} U'_k &:= [u'_1, u'_2, \ldots, u'_k] \in \mathbb{R}^{mr \times k}, \\ V'_k &:= [v'_1, v'_2, \ldots, v'_k] \in \mathbb{R}^{(m+1)r \times k}, \\ \Sigma'_k &:= \mathrm{diag}[\sigma'_1, \sigma'_2, \ldots, \sigma'_k] \in \mathbb{R}^{k \times k}, \end{aligned} \tag{4.3}$$

where U'_k and V'_k have orthonormal columns and

$$B_{mr,(m+1)r} V'_k = U'_k \Sigma'_k, \qquad B^T_{mr,(m+1)r} U'_k = V'_k \Sigma'_k. \tag{4.4}$$

We refer to (4.4) as a partial singular value decomposition of $B_{mr,(m+1)r}$. It now follows from $E_r^T B_{mr} = S^{(m)} E_r^T$ and

$$B_{mr,(m+1)r} B^T_{mr,(m+1)r} = B_{mr} B^T_{mr} + E_r L^{(m+1)} (L^{(m+1)})^T E_r^T \tag{4.5}$$

that $\hat{\theta}_j := (\sigma'_j)^2$ is an eigenvalue and $\hat{w}_j = B^{-1}_{mr} u'_j$ an accompanying eigenvector of (4.1). Thus, $\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_k$ determined in this manner are the k smallest harmonic Ritz values of $A^T A$. The harmonic Ritz vector of $A^T A$ associated with $\hat{\theta}_j$ is defined by

$$\hat{v}_j := P_{mr} \hat{w}_j, \tag{4.6}$$

see, e.g., [1, 14, 15, 17].

Similarly to (3.1), we have

$$A^T A \hat{v}_j - \hat{\theta}_j \hat{v}_j = \bar{P}^{(m+1)} (L^{(m+1)})^T E_r^T u'_j, \qquad 1 \leq j \leq k,$$

where

$$\bar{P}^{(m+1)} := P^{(m+1)} - P_{mr} B^{-1}_{mr} E_r L^{(m+1)}, \tag{4.7}$$

i.e., the residual errors for all harmonic Ritz pairs $\{\hat{\theta}_j, \hat{v}_j\}$ live in $\mathcal{R}(\bar{P}^{(m+1)})$.

We turn to the derivation of relations analogous to (3.7) and (3.11). Equations (4.6) and (4.7) yield

$$[\hat{v}_1 \sigma'_1, \hat{v}_2 \sigma'_2, \ldots, \hat{v}_k \sigma'_k, \bar{P}^{(m+1)}] = P_{(m+1)r} \begin{bmatrix} B^{-1}_{mr} U'_k \Sigma'_k & -B^{-1}_{mr} E_r L^{(m+1)} \\ 0 & I_r \end{bmatrix},$$

where the matrix $P_{(m+1)r}$ is the same as in (2.3). Introduce the QR-factorization

$$\begin{bmatrix} B^{-1}_{mr} U'_k \Sigma'_k & -B^{-1}_{mr} E_r L^{(m+1)} \\ 0 & I_r \end{bmatrix} =: Q'_{k+r} R'_{k+r}, \tag{4.8}$$

where $Q'_{k+r} \in \mathbb{R}^{(m+1)r\times(k+r)}$ has orthonormal columns and $R'_{k+r} \in \mathbb{R}^{(k+r)\times(k+r)}$ is upper triangular. The matrix

$$\hat{P}_{k+r} := P_{(m+1)r}Q'_{k+r} \tag{4.9}$$

has orthonormal columns. Application of equations (1.2) and (4.8) shows that

$$\begin{aligned}
A\hat{P}_{k+r} &= [AP_{mr}, AP^{(m+1)}]Q'_{k+r} \\
&= [Q_{mr}B_{mr}, AP^{(m+1)}]\begin{bmatrix} B_{mr}^{-1}U'_k\Sigma'_k & -B_{mr}^{-1}E_rL^{(m+1)} \\ 0 & I_r \end{bmatrix}(R'_{k+r})^{-1} \\
&= [Q_{mr}U'_k\Sigma'_k, AP^{(m+1)} - Q^{(m)}L^{(m+1)}](R'_{k+r})^{-1}.
\end{aligned}$$

Consider the matrix

$$\hat{Q}_k := Q_{mr}U'_k. \tag{4.10}$$

It has orthonormal columns and we compute the Fourier coefficients

$$\hat{C}^{(m+1)} := \hat{Q}_k^T(AP^{(m+1)} - Q^{(m)}L^{(m+1)}) \in \mathbb{R}^{k\times r},$$

as well as the QR-factorization

$$\hat{Q}^{(m+1)}\hat{S}^{(m+1)} := AP^{(m+1)} - Q^{(m)}L^{(m+1)} - \hat{Q}_k\hat{C}^{(m+1)},$$

where $\hat{Q}^{(m+1)} \in \mathbb{R}^{\ell\times r}$ has orthonormal columns and $\hat{S}^{(m+1)} \in \mathbb{R}^{r\times r}$ is upper triangular. Define the matrix

$$\hat{Q}_{k+r} := [\hat{Q}_k, \hat{Q}^{(m+1)}] \in \mathbb{R}^{\ell\times(k+r)}. \tag{4.11}$$

Note that $\hat{Q}_{k+r}$ has orthonormal columns. The matrix

$$\hat{B}_{k+r} := \begin{bmatrix} \sigma'_1 & & 0 & & \\ & \sigma'_2 & & & \hat{C}^{(m+1)} \\ & & \ddots & & \\ & & & \sigma'_k & \\ 0 & & & & \hat{S}^{(m+1)} \end{bmatrix}(R'_{k+r})^{-1} \in \mathbb{R}^{(k+r)\times(k+r)} \tag{4.12}$$

is the product of two upper triangular matrices, one of which has nonzero entries only on the diagonal and in the last r columns. In particular, $\hat{B}_{k+r}$ is upper triangular. We have

$$A\hat{P}_{k+r} = \hat{Q}_{k+r}\hat{B}_{k+r}, \tag{4.13}$$

which is the wanted analogue of (3.7).

We now derive an analogue of the decomposition (3.11). Let $\hat{Q}_k$ be given by (4.10). Equations (2.3) and (4.4) yield

$$A^T\hat{Q}_k = A^TQ_{mr}U'_k = P_{(m+1)r}B^T_{mr,(m+1)r}U'_k = P_{(m+1)r}V'_k\Sigma'_k. \tag{4.14}$$

It follows from the left-hand side decomposition of (4.4) that

$$[I_{mr}, B_{mr}^{-1}E_rL^{(m+1)}]V'_k = B_{mr}^{-1}U'_k\Sigma'_k,$$

and, hence,

$$V_k' = \begin{bmatrix} B_{mr}^{-1} U_k' \Sigma_k' & -B_{mr}^{-1} E_r L^{(m+1)} \\ 0 & I_r \end{bmatrix} \begin{bmatrix} I_k \\ E_r^T V_k' \end{bmatrix}. \tag{4.15}$$

Substituting (4.15) into (4.14), using (4.8) and (4.9), gives

$$A^T \hat{Q}_k = P_{(m+1)r} Q_{k+r}' R_{k+r}' \begin{bmatrix} I_k \\ E_r^T V_k' \end{bmatrix} \Sigma_k' = \hat{P}_{k+r} R_{k+r}' \begin{bmatrix} I_k \\ E_r^T V_k' \end{bmatrix} \Sigma_k'. \tag{4.16}$$

Let $\hat{B}_{k,k+r}$ denote the leading $k \times (k+r)$ submatrix of the upper triangular matrix $\hat{B}_{k+r}$ in (4.13). Then we obtain from (4.13) that

$$\hat{Q}_k^T A \hat{P}_{k+r} = \hat{B}_{k,k+r}. \tag{4.17}$$

It follows from (4.16) that

$$\hat{P}_{k+r}^T A^T \hat{Q}_k = R_{k+r}' \begin{bmatrix} I_k \\ E_r^T V_k' \end{bmatrix} \Sigma_k',$$

and a comparison with (4.17) shows that

$$R_{k+r}' \begin{bmatrix} I_k \\ E_r^T V_k' \end{bmatrix} \Sigma_k' = \hat{B}_{k,k+r}^T.$$

Thus, equation (4.16) can be expressed as

$$A^T \hat{Q}_k = \hat{P}_{k+r} \hat{B}_{k,k+r}^T. \tag{4.18}$$

We turn to the last r columns, $A^T \hat{Q}^{(m+1)}$, of $A^T \hat{Q}_{k+r}$. Equation (4.13) yields

$$\hat{P}_{k+r}^T A^T \hat{Q}^{(m+1)} = \hat{B}_{k+r}^T \hat{Q}_{k+r}^T \hat{Q}^{(m+1)} = \hat{B}_{k+r}^T E_r$$

and, therefore,

$$A^T \hat{Q}^{(m+1)} = \hat{P}_{k+r} \hat{B}_{k+r}^T E_r + \hat{F}_r, \tag{4.19}$$

where $\hat{F}_r \in \mathbb{R}^{n \times r}$ and $\hat{P}_{k+r}^T \hat{F}_r = 0$. Combining (4.18) and (4.19) yields

$$A^T \hat{Q}_{k+r} = \hat{P}_{k+r} \hat{B}_{k+r}^T + \hat{F}_r E_r^T, \tag{4.20}$$

which is the desired analogue of (3.11). Note that the residual matrix $\hat{F}_r$ can be computed from equation (4.19), since the other terms are explicitly known.

Given the decompositions (4.13) and (4.20), we can proceed similarly as in Section 3 to compute the decompositions

$$A \hat{P}_{k+ir} = \hat{Q}_{k+ir} \hat{B}_{k+ir}, \qquad A^T \hat{Q}_{k+ir} = \hat{P}_{k+ir} \hat{B}_{k+ir}^T + \check{F}_r E_r^T, \tag{4.21}$$

which are analogous to (3.18). In particular, $\hat{P}_{k+ir} \in \mathbb{R}^{n \times (k+ir)}$ and $\hat{Q}_{k+ir} \in \mathbb{R}^{\ell \times (k+ir)}$ have orthonormal columns with leading submatrices $\hat{P}_{k+r}$ and $\hat{Q}_{k+r}$, respectively. The matrix $\hat{B}_{k+ir} \in \mathbb{R}^{(k+ir) \times (k+ir)}$ is upper block-bidiagonal with leading principal submatrix $\hat{B}_{k+r}$.

Having determined the decompositions (4.21), we proceed by computing the singular value decomposition of $\hat{B}_{k+ir}$. The k smallest singular triplets of $\hat{B}_{k+ir}$ give us approximations of the k smallest singular triplets of A, cf. (2.11), as well as k new harmonic Ritz vectors (4.6). We continue by augmenting the latter vectors by a block Krylov subspace as described in this section. An algorithm for block-size $r = 1$ is presented in [2, Subsection 3.3].

We remark that the accurate computation of the matrix $B_{mr}^{-1}E_r$, used in (4.8), can be difficult when the matrix B_{mr} has a large condition number $\kappa(B_{mr}) := \sigma_{mr}^{(B_{mr})}/\sigma_1^{(B_{mr})}$. In this case, we switch from augmentation of harmonic Ritz vectors to augmentation of Ritz vectors; see [2, Section 3.3] for details.

5. Numerical examples. We compare the MATLAB function `irlbablk`, which implements the block Lanczos bidiagonalization method of the present paper, to the MATLAB function `irlba`, which implements the Lanczos bidiagonalization method (with block-size 1) described in [2]. The functions `irlbablk` with block-size $r = 1$ and `irlba` yield identical results. MATLAB codes for both functions are available from Netlib[1], where also a primer for their use, a demo with a graphic user interface, as well as code for reproducing the computed examples of this section can be found. All codes have been tested with MATLAB versions 6.5-7.2.

The following examples illustrate that it may be beneficial to use a block-method for certain problems. The execution of `irlbablk` is determined by certain user-specified parameters; see Table 5.1. The parameters for `irlba` can be found in [2, Section 4], where also a comparison of `irlba` with other recently proposed methods is presented.

Both `irlbablk` and `irlba` implement the following reorthogonalization strategies; cf. the discussion on reorthogonalization in Section 2. The codes apply one-sided reorthogonalization when the matrix A is fairly well conditioned (as in Examples 1, 2, and 4) and two-sided reorthogonalization when A is ill-conditioned (as in Example 3). For the examples of this section, one- and two-sided reorthogonalization yield about the same accuracy. We therefore do not report results for both reorthogonalization strategies; a comparison of these strategies for `irlba` can be found in [2].

In the computed examples, we determine the initial block P_r by orthonormalizing the columns of an $n \times r$ matrix with normally distributed random entries. When comparing the performance for different block-sizes, we generate the matrix P_s, with s being the largest block-size in the experiment, and let in the computations with block-size $r \leq s$, P_r consist of the first r columns of P_s. In particular, the initial vector for `irlba` is the first column of P_s.

All computations were carried out in MATLAB version 6.5 R13 on a Dell 530 workstation with two 2.4 GHz (512k cache) Xeon processors and 2 GB (400 MHz) of memory running the Windows XP operating system. Machine epsilon is $\epsilon \approx 2.2 \cdot 10^{-16}$.

Example 1. We consider the computation of the $k = 2$ smallest singular triplets of the diagonal matrix

$$A = \mathrm{diag}[1, 1 + \frac{1}{200^4}, 3, 4, \ldots, 200] \in \mathbb{R}^{200\times200} \tag{5.1}$$

and report the performance of `irlbablk` and `irlba`. Restarting is carried out by augmentation of Ritz vectors. We use block-size $r = 2$ for `irlbablk`.

[1] http://www.netlib.org/numeralgo/na26

TABLE 5.1
Parameters for `irlbablk`.

adjust	Initial number of vectors added to the k restart vectors to speed up convergence. Default value: $adjust = 3$.
aug	A 4-letter string. The value 'RITZ' yields the augmentation described in section 3; the value 'HARM' gives augmentation according to section 4. Default value: aug ='HARM' if $sigma$ ='SS', and aug ='RITZ' if $sigma$ ='LS'.
blsz	Block-size of block Lanczos bidiagonal matrix. The parameter specifies the value of r in (1.2)-(1.3). Default value: $blsz = 3$.
disps	When $disps > 0$, available approximations of the k desired singular values and norms of associated residual errors are displayed each iteration; $disps = 0$ inhibits display of these quantities. Default value: $disps = 0$.
k	Number of desired singular triples. Default value: $k = 6$.
maxit	Maximum number of restarts. Default value: $maxit = 1000$.
m	Maximum number of consecutive block Lanczos bidiagonalization steps in the beginning of the computations. This parameter specifies the largest value of m in (1.2)-(1.3) and determines the storage requirement of the method. The code may increase the value of m during execution to speed up convergence, in which case a warning message is displayed. Default value: $m = 10$.
reorth	A 3-letter string. The value 'ONE' yields one-sided full reorthogonalization of the "shorter" vectors; the value 'TWO' gives two-sided full reorthogonalization. When our available estimate of $\kappa(A)$, see the discussion following (2.13), is larger than $\epsilon^{-1/2}$, two-sided full reorthogonalization is used. Default value: $reorth$ ='ONE'.
sigma	A 2-letter string ('SS' for smallest and 'LS' for largest) which specifies which extreme singular triplets are to be computed. Default value: $sigma$ ='LS'.
tol	Tolerance used for convergence check, same as δ in (2.13). Default value: $tol = 10^{-6}$.
P_r	Initial matrix of r columns for the block Lanczos bidiagonalization. When $\ell \geq n$, $P^{(1)} := P_r$; cf. Algorithm 2.1. Default value: P_r is an $n \times r$ matrix with orthonormal columns obtained by QR-factorization of an $n \times r$ matrix with normally distributed random entries.

The function `irlbablk` carries out m block Lanczos steps before restart. We let $m = 10$ or $m = 20$, and let $i = m - 1$ in (3.18). The main storage requirement is for the matrices P_{mr} and Q_{mr} generated before the first restart, as well as for the matrices $\tilde{P}_{k+ir}$ and $\tilde{Q}_{k+ir}$ in (3.18) obtained after restarting. The latter matrices over-write P_{mr} and Q_{mr}. Thus, the storage requirement for the matrices P_{mr}, Q_{mr}, $\tilde{P}_{k+ir}$, and $\tilde{Q}_{k+ir}$ is $4m$ n-vectors.

The function `irlba` is called with $m = 20$ and the storage requirement for the corresponding matrices is $2m$ n-vectors. Thus, `irlbablk` with $m = 10$ requires about the same amount of storage as `irlba` with $m = 20$.

The top graphs of Figure 5.1 show the performance of `irlbablk` with $m = 10$ (solid graphs) and `irlba` with $m = 20$ (dotted graphs). The solid graph labeled

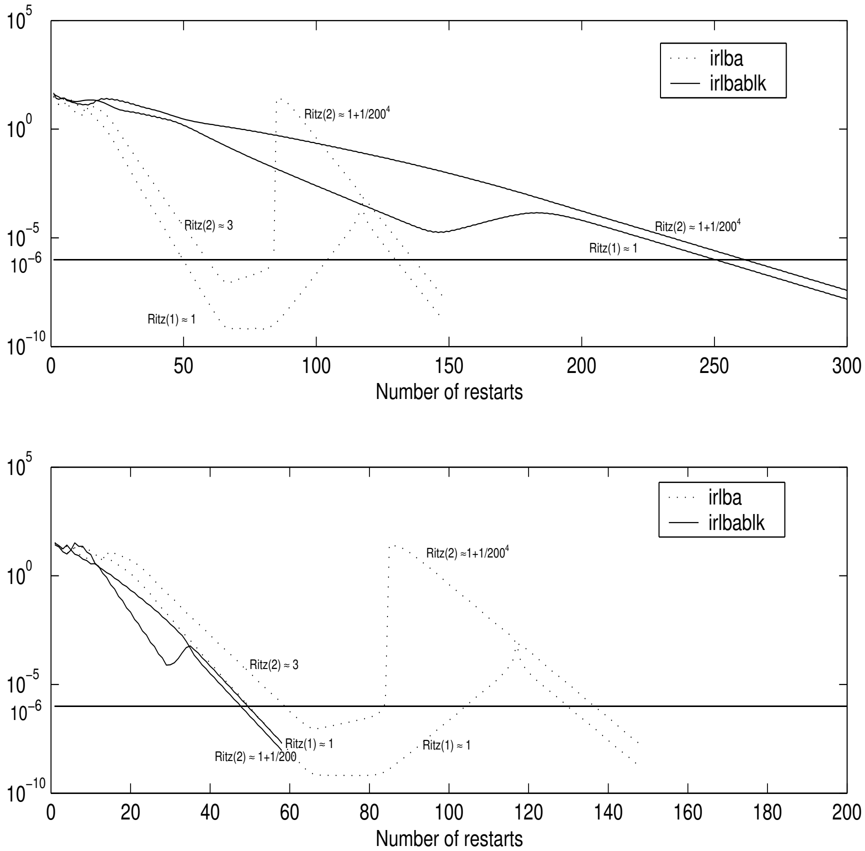

FIG. 5.1. *Example 1: Approximations of the* 2 *smallest singular triplets of the matrix (5.1). For the top graphs both* `irlbablk` *(with block-size $r = 2$) and* `irlba` *use about the same amount of storage. For the bottom graphs the storage for* `irlbablk` *is doubled. This increases the rate of convergence.* `irlba` *has difficulties determining the second smallest singular value.*

Ritz(1) $\approx$ 1 displays the norm of the residual error for the smallest Ritz value, cf. (2.13). This Ritz value approximates the singular value 1. The graph labeled Ritz(2) $\approx 1 + 1/200^4$ displays the norm of the residual error for the second smallest Ritz value, which approximates $1 + 1/200^4$. As can be expected, the residual error associated with the smallest Ritz value is smaller than the residual error associated with the second smallest Ritz value.

The solid horizontal line marks the tolerance $\delta = 1 \cdot 10^{-6}$. Thus, if this value of δ is chosen in (2.13), then `irlbablk` requires about 260 restarts to determine approximations of the 2 smallest singular values with desired accuracy.

The dotted graph labeled Ritz(1) ≈ 1 shows the norm of the residual error for the smallest Ritz value determined by `irlba`. This Ritz value approximates the singular value 1. The code `irlba` requires fewer restarts than `irlbablk` to determine an approximation of the smallest singular value of A with an associated residual error of $1 \cdot 10^{-6}$ because the value of m used for `irlba` is twice as large as for `irlbablk`. However, `irlba` fails to find the second smallest singular value, $1 + 1/200^4$, within the first 80 restarts. The dotted curve labeled Ritz(2) ≈ 3 displays the norm of the residual error for the second smallest Ritz value determined by `irlba`. Until about 85 restarts, this Ritz value approximates the third smallest singular value, 3, of A.

The stopping criterion for `irlba` is (2.13) with $r = 1$. Let $\delta = 1 \cdot 10^{-6}$ in this stopping criterion. Then `irlba` computes approximations of the singular values 1 and 3 to specified accuracy within about 60 restarts. Since the stopping criterion is satisfied the computations are terminated at this point. There is no indication to a user of the code that the computed Ritz values do not approximate the 2 smallest singular values of A.

Such an indication is not delivered until about 85 restarts, when the residual errors increase and `irlba` determines an approximation of the singular values 1 and $1 + 1/200^4$. Note that the residual errors for both Ritz values increase when `irlba` "discovers" the singular value $1 + 1/200^4$.

This example illustrates that if there are close singular values and it is important to determine all the associated singular triplets, then `irlbablk` does this more reliably than `irlba`.

The rate of convergence of `irlbablk` displayed in the top graphs of Figure 5.1 can be increased by choosing a larger value of m. Results for $m = 20$ are shown on the bottom part of Figure 5.1. The number of restarts, as well as the total computational work, is reduced by increasing m to 20, however, the required computer storage is increased.

We finally remark that the behavior `irlbablk` and `irlba` is similar if instead of Ritz vectors, harmonic Ritz vectors are augmented at restart. In particular, `irlba` also misconverges in this situation. We therefore do not report the details of the computations with harmonic Ritz vectors. □

Example 2. Let $A \in \mathbb{R}^{324 \times 324}$ be obtained by discretizing the 2-dimensional negative Laplace operator on the unit square by the standard 5-point stencil with Dirichlet boundary conditions. The MATLAB command

$$A = \mathtt{delsq}(\mathtt{numgrid}('\mathtt{S}', 20)) \tag{5.2}$$

determines this matrix. We would like to compute the 6 largest singular values of A to high accuracy and let δ be machine epsilon in (2.13). Restarting is carried out by augmentation of Ritz vectors. The dominating storage requirement for `irlbablk`, already discussed in Example 1, is proportional to mr, where as usual r is the block-size and m is the largest number of consecutive block Lanczos steps. We limit the memory requirement of `irlbablk` by choosing m and r so that $mr \leq 20$. Thus, we applied `irlbablk` with the $\{r, m\}$ pairs $\{1, 20\}, \{2, 10\}, \{3, 6\}, \{4, 5\}, \{5, 4\}$.

We refer to the computations between each consecutive restart as an iteration. The top histogram of Figure 5.2 displays the average CPU-time for one iteration using several different block-sizes r. The figure shows that the CPU-time per iteration is smaller for block-sizes $r > 1$ than for block-size $r = 1$. This depends on more efficient memory management when $r > 1$.

The bottom histogram of Figure 5.2 shows the total CPU-time required to determine the 6 largest singular triplets of A with desired accuracy, as well as the total number of matrix-vector products (mvp) with the matrices A and A^T. Here the evaluation of Aw with $w \in \mathbb{R}^{324 \times r}$ counts as r matrix-vector products. The smallest total CPU-time is achieved for block-size $r = 2$, even though the smallest number of matrix-vector products is obtained for $r = 1$. This example illustrates that block-methods may require less total CPU-time than methods that work with single vectors (block-size $r = 1$). The codes `irlbablk` for block-size $r = 1$ and `irlba` were found to

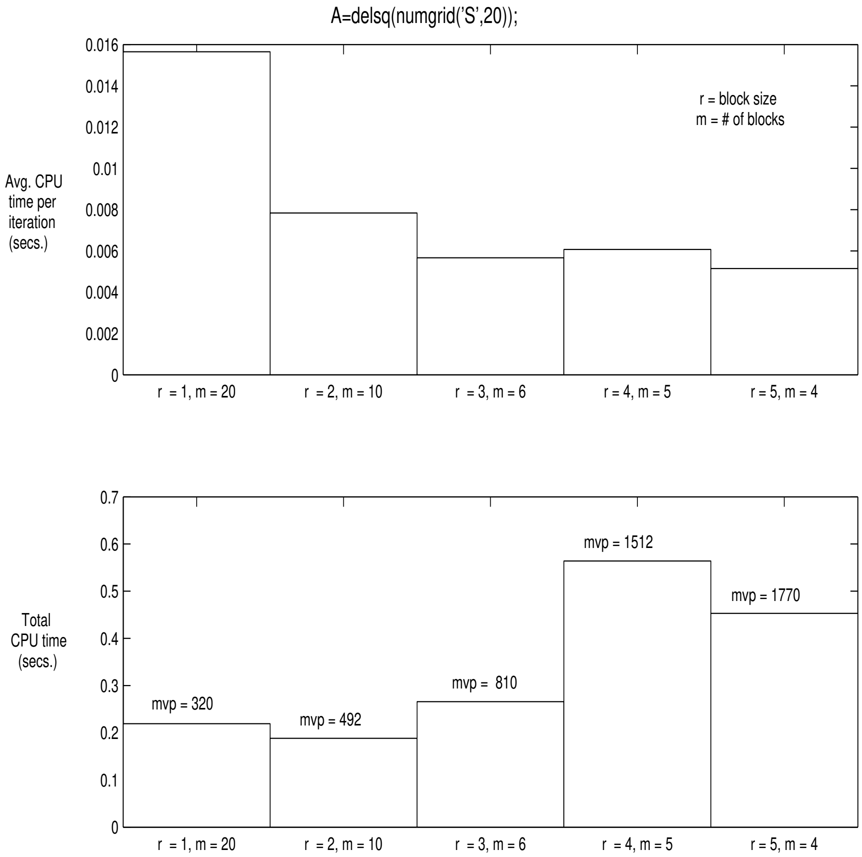

FIG. 5.2. *Example 2: Computation of the* 6 *largest singular triplets of the matrix (5.2). The top histogram shows the average CPU-time between consecutive restarts. The bottom histogram displays the total CPU-time required and the number of matrix-vector product evaluations needed to compute the desired singular triplets to desired accuracy.*

require essentially the same amount of CPU-time. We therefore only report timings for the former.

We remark that the combination of r and m that requires the least total CPU-time depends on the problem at hand and on the architecture of the computer used. For instance, suppose that there are no singular value-clusters, and assume that the computer architecture and matrix-storage format are such that the evaluation of matrix-vector products with a block-vector with r columns takes r times as long as the sequential evaluation of matrix-vector products with r (single) vectors. Then, block-size $r = 1$ will give the shortest CPU-time. Under different circumstances, the CPU-time may be smaller for a block-size $r > 1$; see also the discussion on the use of sparse BLAS towards the end of Section 1. For many problems, the main advantage of using block-size $r > 1$ is increased reliability. $\square$

Example 3. We would like to determine the 4 smallest singular triplets of the

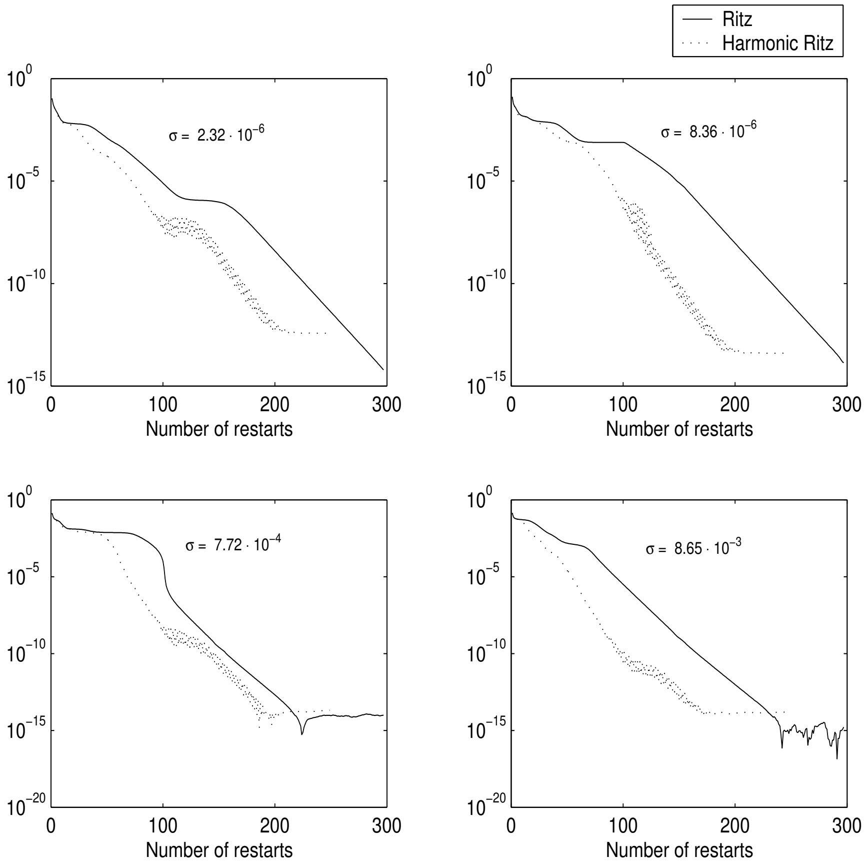

FIG. 5.3. *Example 3: Graphs showing the errors in computed approximations of the* 4 *smallest singular values of the matrix (5.3). The solid graphs show the errors obtained by augmentation of Ritz vectors and the dotted graphs shows the errors obtained by augmentation of harmonic Ritz vectors.*

symmetric Toeplitz matrix

$$T = \begin{bmatrix} t_1 & t_2 & & \cdots & t_{n-1} & t_n \\ t_2 & t_1 & t_2 & & & t_{n-1} \\ & t_2 & t_1 & & & \\ \vdots & \vdots & & \ddots & & \vdots \\ & & & & & \\ t_{n-1} & & & & & t_2 \\ t_n & t_{n-1} & & \cdots & t_2 & t_1 \end{bmatrix} \in \mathbb{R}^{n\times n} \tag{5.3}$$

with $n = 130$ and

$$t_i = \begin{cases} 1, & \text{if } i = 1, \\ \left(\dfrac{\mu \sin(\frac{i-1}{\mu})}{(i-1)} \right)^2, & \text{if } i = 2, 3, \ldots, 8, \\ 0, & \text{otherwise.} \end{cases}$$

This matrix has previously been considered by Luk and Qiao [13], who determine its rank, as well as by Nagy [16]. The 4 smallest singular values are about $\sigma_1 = 2.32 \cdot 10^{-6}$, $\sigma_2 = 8.36 \cdot 10^{-6}$, $\sigma_3 = 7.72 \cdot 10^{-4}$, and $\sigma_4 = 8.65 \cdot 10^{-3}$. Figure 5.3 shows the errors in the approximations of these singular values determined by `irlbablk` with block-size $r = 4$ and $m = 10$ as a function of the number of restarts. The solid graphs shows the errors in the computed approximate singular values determined when restarting is carried out by augmentation of Ritz vectors and the dotted graphs display the errors in the approximations determined when restarting by augmentation of harmonic Ritz vectors is used. The latter approach is seen to yield the smallest errors in all of the desired singular values at almost all restarts. We remark that an analogous example for block-size 1 (using the code `irlba`) is presented in [2]. □

TABLE 5.2
Example 4: Computation of the 10 largest singular triplets of a term-by-document matrix.

block-size	# of blocks	# mat.-vec. products	# matrix access	CPU time: mat.-vec. products	CPU time: other computations	CPU time: total
$r = 1$	$m = 20$	80	80	0.359s	0.500s	0.859s
$r = 2$	$m = 10$	104	52	0.357s	0.471s	0.828s
$r = 3$	$m = 7$	162	54	0.408s	0.717s	1.125s
$r = 4$	$m = 5$	248	62	0.606s	0.941s	1.547s

Example 4. A common task in information retrieval is the computation of a few of the largest singular triplets of a term-by-document matrix. These matrices can be very large and furnish important examples of problems that require out-of-core memory access to evaluate matrix-vector products. The largest singular triplets determine a low-rank approximation of the term-by-document matrix and the angles between the search vectors and the columns of the computed low-rank approximation are used for informational retrieval; see, e.g., Berry et al. [3] for further details.

The access to matrices that have to be stored out-of-core is very CPU-time demanding, and therefore it typically is advantageous to compute more than one matrix-vector product for each matrix access.

In the present example we consider the term-by-document matrix HYPATIA, which is included in the package *na26* of Netlib associated with the present paper. HYPATIA is of size 11390×1265 and has 109056 non-zero terms from the web server of the Department of Mathematics at the University of Rhode Island. HYPATIA was created in the same manner as many standard term-by-document test matrices; see the TMG web page[2] for details.

We seek to determine the 10 largest singular triplets of HYPATIA with `irlbablk` and limit the memory requirement by choosing the parameters m and r so that $mr \leq 21$. Thus, we use `irlbablk` with the $\{r, m\}$ pairs $\{1, 20\}, \{2, 10\}, \{3, 7\}, \{4, 5\}$. Restarts are carried out by augmentation of Ritz vectors.

Table 5.2 displays the performance of `irlbablk`. The table shows the number of matrix accesses to decrease as the block-size r increases from 1 to 3. The least total CPU-time and the least number of matrix accesses are achieved for block-size $r = 2$, even though the least number of matrix-vector product evaluations is required when $r = 1$, where the number of matrix-vector product evaluations is counted as in Example 2. □

[2] http://scgroup.hpclab.ceid.upatras.gr/scgroup/Projects/TMG

6. Conclusion. This paper presents a block-generalization of the restarted Lanczos bidiagonalization method described in [2]. Computed examples illustrate that the block-method implemented by the MATLAB function `irlbablk` can determine desired singular triplets more reliably than the code `irlba` presented in [2] when the associated singular values are very close. Moreover, in certain situations `irlbablk` may determine desired singular triplets to specified accuracy faster than `irlba`.

Acknowledgement. We would like to thank Michela Redivo-Zaglia for comments.

REFERENCES

[1] J. Baglama, D. Calvetti, and L. Reichel, *IRBL: An implicitly restarted block Lanczos method for large-scale Hermitian eigenproblems*, SIAM J. Sci. Comput., 24 (2003), pp. 1650–1677.

[2] J. Baglama and L. Reichel, *Augmented implicitly restarted Lanczos bidiagonalization methods*, SIAM J. Sci. Comput., 27 (2005), pp. 19–42.

[3] M. W. Berry, S. T. Dumais, and G. W. O'Brien, *Using linear algebra for intelligent information retrieval*, SIAM Review, 37 (1995), pp. 573–595.

[4] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.

[5] Å. Björck, E. Grimme, and P. Van Dooren, *An implicit shift bidiagonalization algorithm for ill-posed systems*, BIT, 34 (1994), pp. 510–534.

[6] G. H. Golub, F. T. Luk, and M. L. Overton, *A block Lanczos method for computing the singular values and corresponding vectors of a matrix*, ACM Trans. Math. Software, 7 (1981), pp. 149–169.

[7] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, 1996.

[8] M. E. Hochstenbach, *A Jacobi-Davidson type SVD method*, SIAM J. Sci. Comput., 23 (2001), pp. 606–628.

[9] M. E. Hochstenbach, *Harmonic and refined extraction methods for the singular value problem, with applications in least-squares problems*, BIT, 44 (2004), pp. 721–754.

[10] Z. Jia and D. Niu, *An implicitly restarted refined bidiagonalization Lanczos method for computing a partial singular value decomposition*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 246–265.

[11] E. Kokiopoulou, C. Bekas, and E. Gallopoulos, *Computing smallest singular triplets with implicitly restarted Lanczos bidiagonalization*, Appl. Numer. Math., 49 (2004), pp. 39–61.

[12] R. B. Lehoucq, *Implicitly restarted Arnoldi methods and subspace iteration*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 551–562.

[13] F. T. Luk and S. Qiao, *Rank-revealing decomposition of symmetric Toeplitz matrices*, in Advanced Signal Processing Algorithms, ed. F. T. Luk, Proc. SPIE, vol. 2563, 1995, pp. 293–301.

[14] R. B. Morgan, *Computing interior eigenvalues of large matrices*, Linear Algebra Appl., 154-156 (1991), pp. 289–309.

[15] R. B. Morgan, *Restarted block GMRES with deflation of eigenvalues*, Appl. Numer. Math., 54 (2005), pp. 222–236.

[16] J. G. Nagy, *Fast algorithms for the regularization of banded Toeplitz least squares problems*, in Advanced Signal Processing Algorithms, Architectures, and Implementations IV, ed. F. T. Luk, Proc. SPIE, vol. 2295, 1994, pp. 566–575.

[17] C. C. Paige, B. N. Parlett, and H. A. van der Vorst, *Approximate solutions and eigenvalue bounds from Krylov subspaces*, Numer. Linear Algebra Appl., 2 (1995), pp. 115–134.

[18] B. Parlett, *Misconvergence in the Lanczos algorithm*, in Reliable Numerical Computation, eds. M. G. Cox and S. Hammarling, Clarendon Press, Oxford, 1990, pp. 7–24.

[19] H. D. Simon and H. Zha, *Low rank matrix approximation using the Lanczos bidiagonalization process with applications*, SIAM J. Sci. Comput., 21 (2000), pp. 2257–2274.

[20] D. C. Sorensen, *Numerical methods for large eigenvalue problems*, Acta Numerica, 11 (2002), pp. 519–584.

[21] R. Vudoc, E.-J. Im, and K. A. Yellick, *SPARSITY: Optimization framework for sparse matrix kernels*, Int'l. J. High-Performance Comput. Appl., 18 (2004), pp. 135–158.

[22] H. A. G. Wijshoff, *Implementing sparse BLAS primitives on concurrent/vector processors: a case study*, in Lectrures in Parallel Computation, eds. A. Gibbons and P. Spirakis, Cambridge University Press, Cambridge, 1993, pp. 405–437.

[23] K. Wu and H. Simon, *Thick-restarted Lanczos method for large symmetric eigenvalue problems*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 602–616.