BIT
200x, Vol. xx, No. x, pp. 001–012 DRAFT

BLOCK BIDIAGONAL DECOMPOSITION AND LEAST SQUARES PROBLEMS WITH MULTIPLE RIGHT-HAND SIDES

ÅKE BJÖRCK

Department of Mathematics, University of Linköping
S-581 83 Linköping, Sweden. email: akbjo@math.liu.se

Abstract.

The bidiagonal decomposition of a matrix A plays an important role in algorithms for computing the SVD and also for solving least squares and total least squares problems. In a seminal paper from 1965 Golub and Kahan gave two different methods for computing this decomposition. The first uses Householder transformations applied alternately from right and left. The second is based on a Lanczos process and forms the core of the LSQR algorithm for sparse least squares problems.

In this paper we first develop generalizations of the Lanczos and Householder algorithms to the case of block bidiagonalization. We then show how to apply these algorithms to solving least squares and total least squares problems with multiple right hand sides. The resulting methods are interpreted as projection methods onto certain block Krylov subspaces. We consider also the deflation process needed when rank deficient blocks occur during the decomposition. Applications to partial least squares (PLS) regression with several dependent variables will be mentioned.

AMS subject classification: 65F10, 65F20, 65F25.

Key words: Bidiagonalization, Householder transformations, LSQR, Krylov subspaces, Partial Least Squares.

1 Introduction

In the seminal paper [5] Golub and Kahan gave two algorithms for computing the bidiagonal decomposition (BDD)

$$A = U \begin{pmatrix} B \\ 0 \end{pmatrix} V^T, \quad B = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_n & \alpha_n \\ & & & \beta_{n+1} \end{pmatrix}, \tag{1.1}$$

of a rectangular matrix $A \in \mathbb{R}^{m\times n}$, $m > n$, Here B is a lower bidiagonal matrix and $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are square orthogonal matrices. (The assumption that $m > n$ is for notational convenience only.) In the first algorithm the matrix A is alternately multiplied from the left and right with Householder transformations. The second algorithm uses a Lanczos process.

In [5] the decomposition (1.1) was used as a preliminary step in computing the singular value decomposition of A. In the LSQR algorithm by Paige and Saunders [12] used the Lanczos bidiagonalization algorithm to develop a Krylov subspace method for solving large sparse least squares problems. The sequence of solutions computed by LSQR have a regularization effect and often give better approximate solutions for ill-posed problems than truncated SVD solutions. Therefore it is of interest to note that LSQR can use Householder bidiagonalization. This is a better choice for problems which are amenable to direct matrix factorizations.

A frequently encountered problem in science and engineering is to predict a set of dependent variables from a set of independent variables. Often the number of predictive variables is large and there may be linear dependencies among them. Partial least squares (PLS) regression is a recently much used technique for this purpose. PLS originated in the social sciences but became popular first in chemometrics. For a single right-hand side PLS is mathematically equivalent to LSQR; see [17, 4].

In many applications of scientific computations one needs to solve linear systems or least squares problems with multiple right-hand sides $B = (b_1, \dots, b_p)$. There are two possible approaches; see Gutknecht [9]. In the first a seed right-hand side is first selected. The Krylov subspace generated in the solution of this system is then used to start up the solution of the second system and so on. In the second approach a block Krylov solver is used where all right-hand sides are treated simultaneously. Block Krylov methods have several advantages such as using a much larger search space from the start and introducing matrix–matrix multiplies into the algorithm.

In this paper we develop block Krylov subspace algorithms for solving least squares and total least squares problems with multiple right hand sides. We consider both Householder and Lanczos type algorithms. The similarity and (mathematical) equivalence of these two algorithms is emphasized and used in the development. In Section 2 the bidiagonalization algorithm and its applications to least squares and total least squares problems are briefly reviewed. In Section 3 two block bidiagonalization algorithm are developed. The Lanczos algorithm is essentially equal to that given by Golub, Luk, and Overton [6]. The related Householder block bidiagonalization is also described. In Section 4 we develop algorithms for least squares problems with multiple right hand sides based on block bidiagonalization. The block Lanczos version, which is a natural generalization of the LSQR algorithm, has independently been given by Karimi and Toutounian [11].

One aspect that requires careful consideration in block methods is the deflation needed when rank deficient blocks occur. In the case of a single right-hand side this has been considered in great detail by Paige and Strakoš [13]. This aspect is discussed in Section 5.

2 Bidiagonal Decomposition and Least Squares

In the first algorithm given in [5] the matrix A is alternately multiplied from the left and right with Householder transformations Q_k and P_k, respectively, where Q_1 is chosen so that $Q_1 u_1 = \beta_1 e_1$. Then, for $k = 1:n$, the Householder matrix P_k is chosen to zero the last $n-k$ elements in the kth row of A. Similarly, Q_{k+1} is chosen to zero the last $m-k-1$ elements in the kth column of A. Then for $k = 1:n$, we have

$$Q_{k+1} \cdots Q_2 Q_1 A P_1 \cdots P_k \begin{pmatrix} I_k \\ 0 \end{pmatrix} = \begin{pmatrix} I_{k+1} \\ 0 \end{pmatrix} B_k, \tag{2.1}$$

where

$$B_k = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix} \in \mathbb{R}^{(k+1)\times k}, \tag{2.2}$$

This bidiagonalization algorithm requires $4n^2(m - n/3)$ flops or about twice as much as a QR factorization.

If we set

$$U_k = Q_1 Q_2 \cdots Q_{k+1} \begin{pmatrix} I_{k+1} \\ 0 \end{pmatrix} \in \mathbb{R}^{m\times(k+1)}, \tag{2.3}$$

$$V_k = P_1 P_2 \cdots P_k \begin{pmatrix} I_k \\ 0 \end{pmatrix} \in \mathbb{R}^{n\times k}, \tag{2.4}$$

then from the structure of the Householder matrices Q_k it follows that

$$U \begin{pmatrix} I_{k+1} \\ 0 \end{pmatrix} = Q_1 \cdots Q_k Q_{k+1} \cdots Q_{n+1} \begin{pmatrix} I_{k+1} \\ 0 \end{pmatrix} \tag{2.5}$$

$$= Q_1 \cdots Q_{k+1} \begin{pmatrix} I_{k+1} \\ 0 \end{pmatrix} = U_{k+1}. \tag{2.6}$$

Similarly the first k columns in V_k equal those in V. This shows that in the kth step determines α_k, β_{k+1}, v_k, and u_{k+1}.

The algorithm described above gives a constructive proof of the existence of the decomposition (1.1). It also shows that provided that $\alpha_i \neq 0$, $\beta_{i+1} \neq 0$, $i = 1:n$, the bidiagonal decomposition is uniquely determined once the first column $u_1 = Ue_1$ has been fixed.

In the second algorithm described in [5] for computing the bidiagonal decomposition, the columns in U and V are generated sequentially by a Lanczos process. From (2.1) we immediately get the two relations

$$A(v_1\ v_2 \cdots\ v_n) = (u_1\ u_2 \cdots\ u_{n+1})B \tag{2.7}$$

$$A^T(u_1\ u_2 \cdots\ u_{n+1}) = (v_1\ v_2 \cdots\ v_n)B^T. \tag{2.8}$$

Equating columns in these two equations we obtain $A^T u_1 = \alpha_1 v_1$,

$$\begin{aligned} Av_j &= \alpha_j u_j + \beta_{j+1} u_{j+1}, \quad j = 1:n, \\ A^T u_j &= \beta_j v_{j-1} + \alpha_j v_j, \quad j = 2:n, \end{aligned}$$

and $A^T u_{n+1} = \beta_{n+1} v_n$. Given a unit starting vector u_1 these recurrence relations can be used to compute the vectors $v_1, u_2, v_2, \ldots, u_{n+1}$ in this order. We have $\alpha_1 v_1 = A^T u_1$,

$$\beta_{j+1} u_{j+1} = Av_j - \alpha_j u_j, \quad j = 1:n, \tag{2.9}$$

$$\alpha_j v_j = A^T u_j - \beta_j v_{j-1}, \quad j = 1:n, \tag{2.10}$$

where α_j and β_{j+1} are determined by the condition that $\|v_j\|_2 = \|u_{j+1}\|_2 = 1$. The process can be continued as long as if $\alpha_j \beta_{j+1} \neq 0$. This algorithm is identical to the procedure Bidiag 1 in [12].

The recurrence relations can be written in matrix form as $U_{k+1}(\beta_1 e_1) = b$,

$$AV_k = U_{k+1} B_k, \tag{2.11}$$

$$A^T U_{k+1} = V_k B_k^T + \alpha_{k+1} v_{k+1} e_{k+1}^T. \tag{2.12}$$

From the uniqueness it follows that in exact arithmetic the two algorithms will produce the same bidiagonal decomposition. However, this is not true in finite precision. The Householder algorithm is backward stable and the matrices U and V are by construction orthogonal. As is well known from other applications of the Lanczos process, the Lanczos algorithm suffers from loss of orthogonality and is less stable. The Lanczos algorithm is therefore mainly of interest when A is a large and sparse matrix.

For a square matrix $C \in \mathbb{R}^{n\times n}$ and a vector $z \in \mathbb{R}^n$ define the Krylov subspaces

$$\mathcal{K}_k(C, z) = \operatorname{span}\left\{z, Cz, \ldots, C^{k-1} z\right\}. \tag{2.13}$$

The dimension of $\mathcal{K}_k(C, z)$ will usually be k unless z is specially related to C or $k > n$. From the Lanczos recurrence relations it follows by induction that

$$\operatorname{span}(u_1, \ldots, u_k) = \mathcal{K}_k(AA^T, u_1), \tag{2.14}$$

$$\operatorname{span}(v_1, \ldots, v_k) = \mathcal{K}_k(A^T A, A^T u_1). \tag{2.15}$$

Consider the least squares problem

$$\min_x \|Ax - b\|_2, \quad A \in \mathbf{R}^{m\times n}. \tag{2.16}$$

Often a good approximate solution can be found in a subspace of much smaller dimension than n. The bidiagonal decomposition is the core of the LSQR algorithm of Paige and Saunders [12] for treating problem (2.16). If the unit starting vector in the bidiagonalization is taken to be

$$u_1 = U_k e_1 = (1/\beta_1) b,$$

then $U_k^T b = \beta_1 e_1$. In LSQR, after k steps of bidiagonalization an approximate least squares solutions

$$x_k = V_k y_k \tag{2.17}$$

is determined. By (3.10) this is equivalent to restricting x_k to lie in the Krylov subspace $\mathcal{K}_k(A^T A, A^T b)$. Using (2.11) it follows that

$$b - Ax_k = b - AV_k y_k = U_{k+1}(\beta_1 e_1 - B_k y_k).$$

Using the orthogonality of the columns of U_{k+1} gives

$$\|b - Ax_k\| = \|\beta_1 e_1 - B_k y_k\|.$$

Hence $\|b - Ax_k\|$ is minimized over all $x_k \in \mathcal{R}(V_k)$ by taking y_k as a solution to the bidiagonal least squares problem

$$\min_{y_k} \|\beta_1 e_1 - B_k y_k\|. \tag{2.18}$$

Thus the LSQR algorithm is a projection method, in which an approximate solution in $\mathcal{R}(V_k)$ is determined such that the residual is orthogonal to $\mathcal{R}(U_{k+1})$. The approximations are the optimal solutions on a nested sequence of Krylov subspaces. It follows that the residuals $\|b - Ax_k\|$, $k = 1 : n$, form a non-increasing sequence. The algorithm can be terminated when the norm of the residual has been reduced sufficiently. As done in LSQR, the solution of the sequence of bidiagonal problems can be interleaved with the bidiagonalization. For further details of LSQR we refer to [12].

The least squares solution to (4.5) can be computed from a QR factorization of B_k. This takes the form

$$G_k(B_k \mid \beta_1 e_1) = \left(\begin{array}{c|c} R_k & f_k \\ & \bar{\phi}_{n+1} \end{array} \right) \tag{2.19}$$

where G_k is a product of n Givens rotations. The solution is obtained by back-substitution from $R_k y = d_k$. The norm of the corresponding residual vector equals $|\bar{\phi}_{n+1}|$.

The LSQR algorithm was developed for large and sparse least squares problems and uses the Lanczos algorithm for bidiagonalization. In exact arithmetic the direct algorithm using a sequence of Householder transformations will, at each step, compute the same quantities as as LSQR, which uses the Lanczos process. In LSQR there is a gradual loss of orthogonality as in other algorithms based on the Lanczos process. Although this does not affect the final accuracy it delays convergence; see [3]. Therefore, the backward stable Householder algorithm should be preferred for dense and moderately large problems. Such are often encountered, e.g., in multiple linear regression and in the PLS method.

In the TLS problem the error function to be minimized is

$$\frac{\|b - Ax\|_2^2}{\|x\|_2^2 + 1} = \frac{\|\beta_1 e_1 - By\|_2^2}{\|y\|_2^2 + 1},$$

which is a lower bidiagonal TLS problem. The solution of this bidiagonal TLS problem can be constructed from the right singular vector corresponding to the smallest singular value of the bidiagonal matrix $(\, \beta_1 e_1 \quad B \,)$.

3 Block Bidiagonalization Methods

Let $Q_1 \in \mathbf{R}^{m\times m}$ be a given orthogonal matrix. Then

$$U_1 = Q_1 \begin{pmatrix} I_p \\ 0 \end{pmatrix} \in \mathbf{R}^{m\times p} \tag{3.1}$$

is a matrix with orthogonal columns. Next form $Q_1^T A$ and determine an orthogonal matrix $P_1 \in \mathbf{R}^{n\times n}$ so that the first p rows of $Q_1^T A$ are transformed into lower triangular form

$$(I_p \;\; 0)(Q_1^T A)P_1 = (L_1 \;\; 0).$$

This gives a lower triangular matrix L_1 and a matrix $P_1 E_1 = V_1$ with orthogonal columns. In the block Householder bidiagonalization algorithm we proceed by alternately performing LQ factorizations of blocks of p row and QR factorizations of blocks of p columns of the transformed matrix. After k double steps this algorithm has produced a matrix

$$Q_{k+1}\cdots Q_2 Q_1\, A\, P_k \cdots P_1$$

whose first $(k+1)$ block rows and k block columns equal the block bidiagonal matrix

$$T_k = \begin{pmatrix} L_1 & & & \\ R_2 & L_2 & & \\ & \ddots & \ddots & \\ & & R_k & L_k \\ & & & R_{k+1} \end{pmatrix} \in \mathbf{R}^{(k+1)p\times kp}. \tag{3.2}$$

and orthogonal matrices

$$Q_{k+1}\cdots Q_2 Q_1 \begin{pmatrix} I_{(k+1)p} \\ 0 \end{pmatrix} = (U_1, U_2, \ldots, U_{k+1}), \tag{3.3}$$

$$P_k \cdots P_1 \begin{pmatrix} I_{kp} \\ 0 \end{pmatrix} = (V_1, V_2, \ldots, V_k), \tag{3.4}$$

where $U_j \in \mathbf{R}^{m\times p}$, $V_j \in \mathbf{R}^{n\times p}$. Further, as in the case $p = 1$, it holds that

$$A(V_1\; V_2 \cdots\; V_k) = (U_1\; U_2 \cdots\; U_{k+1})T_k \tag{3.5}$$

$$A^T(U_1\; U_2 \cdots\; U_{k+1}) = (V_1\; V_2 \cdots\; V_k)T_k^T. \tag{3.6}$$

The QR and LQ factorizations can be performed using Householder transformations. Thus we have a constructive proof of the existence of the block bidiagonal decomposition. It also follows that the decomposition is uniquely determined by the first block column U_1. Note that the matrix T_k is a banded *lower* triangular matrix with $(p + 1)$ nonzero diagonals.

The block Lanczos bidiagonalization algorithm, given by Golub, Luk, and Overton in [6], can now be obtained by equating columns in (3.5) and (3.6). This gives

$$AV_j = U_j L_j + U_{j+1} R_{j+1}, \quad j = 1 : n,$$

and $A^T U_1 = V_1 L_1^T$,

$$A^T U_{j+1} = V_j R_{j+1}^T + V_{j+1} L_{j+1}^T, \quad j = 1 : n-1.$$

Given U_1, these recurrence relations can be used to compute the vectors $V_1, U_2, V_2, \ldots, V_n, U_{n+1}$, in this order. Form $A^T U_1$ and compute its QR factorization gives V_1 and L_1^T. Continuing, for $j = 1 : n$, we compute the residual matrix and its QR factorization

$$W_j = AV_j - U_j L_j, \quad U_{j+1} R_{j+1} = W_j. \tag{3.7}$$

giving U_{j+1} and R_{j+1}. Next, compute the residual and its QR factorization

$$Z_{j+1} = A^T U_{j+1} - V_j R_{j+1}^T, \quad V_{j+1} L_{j+1}^T = Z_{j+1}. \tag{3.8}$$

giving V_{j+1} and L_{j+1}. Since the orthogonal vectors are needed explicitly.

We assume here that all residual matrices W_j and Z_{j+1} have full column rank. This is essentially Algorithm 2.1 in [6].

We have assumed here that all QR factorizations yield nonsingular triangular matrices. What action to take when rank deficient blocks arise will be discussed in the next section. From the Lanczos recurrence relations (3.7)–(3.8) it follows by induction that

$$\operatorname{span}(U_1, \ldots, U_k) = \mathcal{K}_k(AA^T, U_1) \tag{3.9}$$

$$\operatorname{span}(V_1, \ldots, V_k) = \mathcal{K}_k(A^T A, A^T U_1). \tag{3.10}$$

This shows that the block bidiagonalization algorithm constructs orthogonal bases for these nested sequence of subspaces of $\mathbf{R}^m$ and $\mathbf{R}^n$, respectively.

4 Least Squares Problems with Multiple Right-Hand Sides

Consider a least squares problem

$$\min_X \|B - AX_k\|_F, \quad B = (u_1, \ldots, u_p) \in \mathbf{R}^{m \times p}. \tag{4.1}$$

with multiple right-hand sides. To get a block version of LSQR we start the block bidiagonalization with the QR factorization

$$Q_1^T B = \begin{pmatrix} R_1 \\ 0 \end{pmatrix}, \quad R_1 \in \mathbf{R}^{p \times p}. \tag{4.2}$$

with R_1 upper triangular. The columns of

$$U_1 = Q_1 \begin{pmatrix} I_p \\ 0 \end{pmatrix}, \tag{4.3}$$

form an orthogonal basis for the column space of B.

A natural generalization of LSQR is to compute a sequence of approximate solutions

$$X_k = V_k Y_k, \quad k = 1, 2, 3, \ldots. \tag{4.4}$$

By (3.10) this is equivalent to restricting X_k to lie in the Krylov subspace $\mathcal{K}_k(A^TA, A^TB)$. Using (2.11) it follows that

$$B - AX_k = B - AV_kY_k = U_{k+1}(R_1E_1 - T_kY_k).$$

Using the orthogonality of the columns of U_{k+1} gives

$$\|B - AX_k\|_F = \|R_1E_1 - T_kY_k\|_F.$$

Hence $\|B - AX_k\|_F$ is minimized over all $X_k \in \mathcal{R}(V_k)$ by taking Y_k as a solution to the bidiagonal least squares problem

$$\min_{Y_k} \|R_1E_1 - T_kY_k\|. \tag{4.5}$$

Thus the LSQR algorithm is a projection method, in which an approximate solution in $\mathcal{R}(V_k)$ is determined such that the residual is orthogonal to $\mathcal{R}(U_{k+1})$. The approximations are the optimal solutions on a nested sequence of Krylov subspaces.

The solution of the LS problems for the p right-hand sides, involves left orthogonal transformations to bring the *lower* triangular banded matrix T_k into *upper* triangular banded form. As in LSQR this can be *interleaved with the block bidiagonalization.* Residual norms are available at each step. For example, at an intermediate step we have

$$\left(\begin{array}{cc|cccc|cccc}
\times & \times & \times & * & * & & & & & \\
* & \times & \otimes & \times & * & * & & & & \\
* & * & \otimes & \otimes & \times & * & * & & & \\
* & * & & \otimes & \otimes & \times & * & * & & \\
\hline
* & * & & & \otimes & \otimes & \times & * & & \\
* & * & & & & \otimes & \times & \times & & \\
& & & & & & \times & \times & \times & \times \\
& & & & & & \times & \times & \times & \times \\
& & & & & & \times & \times & \times & \times \\
& & & & & & \times & \times & \times & \times
\end{array}\right)$$

$$= \begin{pmatrix} C_1 & R_X & S_1 \\ C_2 & 0 & S_2 \end{pmatrix}.$$

At this step the approximate solution to $\min \|AX - B\|_F$, is

$$X = V_k(R_X^{-1}C_1), \qquad \|B - AX\|_F = \|C_2\|_f.$$

The block algorithm can also be used for TLS problems with multiple right-hand sides. Then the error function to be minimized is

$$\frac{\|B - AX\|_2^F}{\|X\|_F^2 + 1} = \frac{\|R_1E_1 - TY\|_2^F}{\|Y\|_F^2 + 1},$$

which is a lower triangular block bidiagonal TLS problem. The solution of this can be constructed from the right singular vectors corresponding to the smallest singular values of the reduced block bidiagonal matrix is needed. For this a reduction to bidiagonal form is required.

5 Rank Deficient Blocks and Deflation

An aspect that requires careful consideration in block methods is the deflation needed when rank deficient blocks occur. We first considerthe case of a single right-hand side. Assume that in Householder bidiagonalization algorithm the computed $\beta_1, \dots, \beta_k$ and $\alpha_1, \dots, \alpha_{k-1}$ are nonzero. If in the next step either $\alpha_k = 0$ or $\beta_{k+1} = 0$, then the process terminates prematurely. The vectors $u_1, u_2, \dots, u_j$ and $v_1, v_2, \dots, v_j$ build orthogonal bases for the first j linearly independent subspaces of the right and left Krylov sequences (2.14)–(2.15). Termination means that the maximal dimensioned Krylov space has been reached in one of these sequences. Deflation is equivalent to *detecting and removing linearly dependent vectors in these sequences.*

As an example, for $k = 3$ in Householder bidiagonalization, the partially reduced matrix has the form

$$Q_3Q_2Q_1 \begin{pmatrix} b & A \end{pmatrix} P_1P_2 = \left(\begin{array}{c|cc|ccc} \beta_2 & \alpha_1 & & & & \\ & \beta_2 & \alpha_2 & & & \\ & & \beta_3 & \times & \otimes & \otimes \\ \hline & & & \times & \times & \times \\ & & & \otimes & \times & \times \\ & & & \otimes & \times & \times \\ & & & \otimes & \times & \times \end{array}\right),$$

If $\alpha_3 = 0$, this means that the right Krylov vectors $A^Tb, (A^TA)A^Tb, (A^TA)^2A^Tb$ are linearly dependent and the maximal dimension equals 2. If $\beta_4 = 0$ the left Krylov vectors $b, (AA^T)b, (AA^T)^2b, (AA^T)^3b$ are linearly dependent and the maximal dimension equals 3.

In general if $\alpha_k = 0$, the reduced matrix has the form

$$U_k^T A V_k = \left(\begin{array}{c|c} B_{k-1} & 0 \\ \hline 0 & A_k \end{array}\right),$$

where $A_k \in \mathbf{R}^{(m-k)\times(n-k+1)}$ and $B_{k-1} \in \mathbf{R}^{k\times(k-1)}$ has full column rank. Setting $x = P_1 \cdots P_{k-1} y = Vy$, leads to the least squares problem

$$\min_y \left\| \begin{pmatrix} B_{k-1} & 0 \\ 0 & A_k \end{pmatrix} \begin{pmatrix} y_1 \\ y \end{pmatrix} - \begin{pmatrix} \beta_1 e_1 \\ 0 \end{pmatrix} \right\|.$$

This decomposes into two independent subproblems:

$$\min_{y_1} \|B_{k-1}y_1 - \beta_1 e_1\|, \qquad \min_{y_2} \|A_k y_2\|.$$

The solution y_1 is uniquely determined and taking $y = 0$ gives the minimum norm solution to the original LS problem. Since B_k has full column rank the solution to the first subproblem is unique. Clearly the minimum norm solution $x = Vy$ is obtained by taking $y_2 = 0$.

If $\beta_{k+1} = 0$, then the reduced matrix has a similar separable form, where now $\hat{B}_k \in \mathbf{R}^{k\times k}$ is *square* and nonsingular. This can only happen if the original system is consistent.

Paige and Strakoš [13] call the subproblem obtained in this way a **core subproblem**. They show that it is the minimally dimensioned subproblem with the following properties:

- The matrix B_k has full column rank and its singular values are simple.
- The right-hand side βe_1 has nonzero components along each left singular vector of B_k.

The first statement follows from the fact the strict separation of eigenvalues of leading principal minors of a tridiagonal matrix with nonzero elements in the outer diagonals; (see Wilkinson [16, § 5.36, p. 299]). The second property is equivalent to saying that all left singular vectors have a nonzero first component (see Wilkinson [16, § 5.48, p.316]).

These properties ensure that any (weighted) TLS subproblem has a unique solution.

We now consider deflation in the block bidiagonalization. We have seen that rank deficiencies are related to linear dependencies in the two sets of Krylov subspaces associated with the bidiagonalization. When using block bidiagonalization for computing singular values as in [6], then a new vector orthogonal to the Krylov subspace generated so far is introduced. The block Lanczos process can then be continued.

When the block bidiagonalization is used for solving a least squares problem purpose then a deflation should be done when a rank deficient block occurs in the LQ or QR factorization. We recall that these factorizations are to be performed without pivoting, since otherwise the band structure would be affected. Thus, a zero element can occur at any place in the diagonal. Then the factorization yields a singular triangular factor that, e.g. can have the form ($p = 4$)

$$\begin{pmatrix} \times & \times & \times & \times \\ & 0 & \times & \times \\ & & 0 & \times \\ & & & 0 \end{pmatrix}.$$

This shows that when a zero diagonal element occurs, all the following elements in the diagonal can be made zero by choosing the Householder transformations appropriately. Since the diagonal in the triangular factors correspond to the *outermost* diagonals in T_k, the bandwidth of T_k will be reduced by one.

For example let $p = 2$, and suppose that the first element in the diagonal of the block R_2 becomes zero.

$$\left(\begin{array}{cc|cccc|cccc}
\times & \times & \times & & & & & & & \\
 & \times & \times & \times & & & & & & \\
 & & \times & \times & \times & & & & & \\
 & & & \otimes & \times & \times & & & & \\
 & & & & \otimes & \times & \otimes & & & \\
\hline
 & & & & & \otimes & \times & \times & \times & \times \\
 & & & & & & \times & \times & \times & \times \\
 & & & & & & \times & \times & \times & \times \\
 & & & & & & \times & \times & \times & \times \\
 & & & & & & \times & \times & \times & \times \\
 & & & & & & \times & \times & \times & \times
\end{array}\right).$$

This will not make the least squares problem decompose, and the reduction has to continue. However, if later another diagonal element in L_3 (say) becomes zero the bandwidth is further reduced by one. At this point the problem decomposes and the reduction terminates. In general, every time a diagonal element becomes zero the bandwidth (and the length of the Lanczos recursions) is reduced by one. When the bandwidth has been reduced p times the process terminates with a separable problem.

Deflation can be interpreted in terms of linear dependencies in the two sets of Krylov subspaces associated with the bidiagonalization. When, for some j, $1 \le j \le p$, a left Krylov vector $(AA^T)^k b_j$ is linearly dependent on previous vectors, then all left and right Krylov vectors of the form

$$(AA^T)^p b_j, \qquad (A^TA)^p A^T b_j, \quad p \ge k,$$

should be removed. Similarly, when a right Krylov vector $(A^TA)^k A^T b_j$ is linearly dependent on previous vectors, then all vectors of the form

$$(A^TA)^p A^T b_j, \qquad (AA^T)^{p+1} b_j, \quad p \ge k,$$

from the right and left sequence, respectively, should be removed. It can be verified that this is achieved by reducing the bandwidth in the Lanczos recurrence.

When the bandwidth has been reduced p times then the left and right Krylov subspaces have both reached their maximal dimensions. The problem is now separable.

When rank deficient blocks occur the Lanczos block bidiagonalization is modified in a way similar to the Householder algorithm. Recall the block Lanczos recurrences (3.7)–(3.8). Suppose that the first rank deficient block is

$$L_k \in \mathbf{R}^{p\times r}, \quad r < p.$$

Then V_k is $n \times r$, i.e. only $r < p$ vectors are determined. In the next step

$$AV_k - U_k L_k \in \mathbf{R}^{m\times r}.$$

Recurrence still works! Block size has been reduced to r. The modification when a rank deficient block R_k occurs is analogous. Note that this analysis covers also the first step, which is the QR factorization of the right-hand side matrix B.

6 Summary

We have developed block bidiagonalization algorithms which are geberalizations of two algorithms given in [5]. The similarity and (mathematical) equivalence of the Householder and Lanczos algorithms has been emphasized and used in the development. We the considered applications to least squares and total least squares problems with $p > 1$ right-hand sides. It was shown how to modify the algorithms when rank deficienies occur.

Dense least squares problems often occur in data analysis and statistical applications. For these the use of the Householder algorithm is recommended because of its backward stability. Contrary to this recommendation, Lanczos like implementations are used in current software for PLS; see [4]. We note that today even quite large dense problems can be handled in seconds on a desktop computer.

The relationship between PLS for several dependent variables and the block LSQR algorithm should be further investigated.

7 Acknowledgements

It is a pleasure to thank Chris Paige and Michael Saunders for valuable comments. In particular Michael Saunders pointed out the reference [11].

REFERENCES

1. Å. Björck. *Numerical Methods for Least Squares Problems.* SIAM, Philadelphia, 1996.
2. Å. Björck. *The calculation of least squares problems. Acta Numerica*, 13:1–53, 2004.
3. Å. Björck, T. Elfving, and Z. Strakoš. Stability of conjugate gradient and Lanczos methods for linear least squares problems. *SIAM J. Matrix Anal. Appl.*, 19:3:720–736, 1998.
4. L. Eldén (2004), Partial least squares vs Lanczos bidiagonalization, I: Analysis of a projection method for multiple projection, *Comp. Stat. Data Anal.* **46:1**, 11–31.
5. G. H. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *SIAM J. Numer. Anal. Ser. B*, 2:205–224, 1965.
6. G. H. Golub, F. T. Luk, and M. L. Overton. A block Lanczos method for computing the singular values and corresponding singular vectors of a matrix. *ACM Trans. Math. Software*, 7:149–169, 1981.
7. G. H. Golub and R. Underwood. The block Lanczos method for computing eigenvalues. In *Mathematical Software III.* J. R. Rice (Ed.), Academic Press, New York, 1977, 361–377.
8. G. H. Golub and C. F. Van Loan. *Matrix Computations,* 3rd ed. Johns Hopkins University Press, Baltimore, 1996.

9. Martin H. Gutknecht. Block Krylov space methods for linear systems with multiple right hand-sides: An introduction. In A. H. Siddiqi, I. S. Duff, and O. Christensen, editors, *Modern Mathematical Models, Methods, and Algorithms for Real World Systems*, pages 420–447. Anarnaya Publishers, New Dehli, India, 2007.

10. M. Hanke. On Lanczos methods for the regularization of discrete ill-posed problems. *BIT*, 41:1008–1018, 2001.

11. S. Karimi and F. Toutounian. The block least squares method for solving nonsymmetric linear systems with multiple right-hand sides. *Appl. Math. Comput.*, 177:852–862, 2006.

12. C. C. Paige and M. A. Saunders. LSQR. An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Software*, 8:43–71, 1982.

13. C. C. Paige and Z. Strakoš. Core problems in linear algebraic systems. *SIAM J. Matrix. Anal. Appl.*, 2005.

14. Axel Ruhe. Implementation aspects of band Lanczos algorithms for computation of eigenvalues of large sparse symmetric matrices. *Math. Comput.*, 146:680–687, 1979.

15. Diana Maria Sima. *Regularization Techniques in Model Fitting and Parameter Estimation.* Ph.D. thesis, Katholieke Universiteit Leuven, Leuven, Belgium, 2006.

16. J. H. Wilkinson. *The Algebraic Eigenvalue Problem.* Clarendon Press, Oxford, UK, 1965.

17. S. Wold, A. Ruhe, H. Wold, and W. J. Dunn. The collinearity problem in linear regression, the partial least squares (PLS) approach to generalized inverses. *SIAM J. Sci. Statist. Comput.*, 5:735–743, 1984.