BIT 34 (1994), 000–000.

AN IMPLICIT SHIFT BIDIAGONALIZATION ALGORITHM FOR ILL-POSED SYSTEMS

ÅKE BJÖRCK, ERIC GRIMME and PAUL VAN DOOREN *†

Department of Mathematics, Linköping University S-581 83, Linköping, Sweden

Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, 1308 W. Main Street, Urbana, IL 61801

Abstract.

Iterative methods based on Lanczos bidiagonalization with full reorthogonalization (LBDR) are considered for solving large scale discrete ill-posed linear least squares problems of the form $\min_x \|Ax-b\|_2$. Methods for regularization in the Krylov subspaces are discussed which use generalized cross validation (GCV) for determining the regularization parameter. These methods have the advantage that no a priori information about the noise level is required. To improve convergence of the Lanczos process we apply a variant of the implicitly restarted Lanczos algorithm by Sorenson using zero shifts. Although this restarted method simply corresponds to using LBDR with a starting vector $(AA^T)^p b$, it is shown that carrying out the process implicitly is essential for numerical stability. An LBDR algorithm is presented which incorporates implicit restarts to ensure that the global minimum of the CGV curve corresponds to a minimum on the curve for the truncated SVD solution. Numerical results are given comparing the performance of this algorithm with non-restarted LBDR.

AMS subject classifications: 65F20.

Key words: Ill-posed problems, Lanczos algorithm, regularization, least squares.

1 Introduction

In this paper we consider reconstruction of the solution $\bar{x}$ to the linear least squares problem

$$\min_{\bar{x}} \|A\bar{x} - \bar{b}\|_2, \qquad A \in R^{m\times n}, \quad m \geq n, \tag{1.1}$$

from noisy data $b = \bar{b} + \epsilon r$, where $\epsilon > 0$ and r is a random noise vector. Such problems arise from discretizations of ill-posed problems in many applications, e.g., in atmospheric studies, geophysics, and profile inversions, see [12].

*This work was partially supported by DARPA under grant 60NANB2D1272 and by the National Science Foundation under grant CCR-9209349

†Received

Let the singular value decomposition of A be

$$A = U_\sigma \Sigma V_\sigma^T = \sum_{i=1}^{n} \sigma_i \, u_{\sigma i} v_{\sigma i}^T, \tag{1.2}$$

where $\sigma_i > 0$, $i = 1, \ldots, n$, and $u_{\sigma i}$ and $v_{\sigma i}$ are the left and right singular vectors of A. Then an approximate solution to (1.1) can be expressed as a singular value expansion

$$x = \sum_{i=1}^{n} \frac{u_{\sigma i}^T b}{\sigma_i} v_{\sigma i}. \tag{1.3}$$

For an ill-posed problem, both the singular values, σ_i, and the right hand side (rhs) coefficients, $u_{\sigma i}^T \bar{b}$, gradually decay towards zero. However, since the random noise vector r tends to have a constant projection along all singular vectors $u_{\sigma i}$, the coefficients corresponding to the perturbed right hand side, $u_{\sigma i}^T b$, tend to level off at some point where

$$u_{\sigma i}^T b = u_{\sigma i}^T(\bar{b} + \epsilon r) \approx \epsilon u_{\sigma i}^T r.$$

For the exact right hand side, the rhs coefficients converge to zero. But the rhs coefficients level off in the presence of noise so that taking more terms in the expansion will make the solution blow up, see Figure 1.1. Hence an approximate solution must

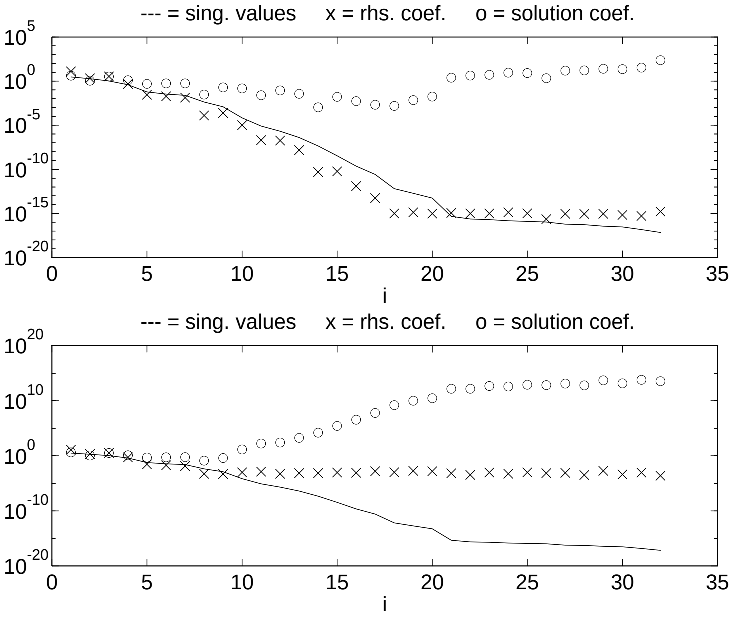

Figure 1.1: Singular values σ_i, rhs coefficients $u_{\sigma i}^T b$, and solution coefficients $(u_{\sigma i}^T b)/\sigma_i$, for the test problem shaw(32) from [14]. In the top plot, the variance of the noise is $\epsilon = 0$; in the lower plot, 10^{-3}.

incorporate some sort of regularization such that the contributions from the noise is damped out. This regularization can be achieved by truncating the expansion (1.3);

$$x_\nu = \sum_{i=1}^{\nu} \frac{u_{\sigma i}^T b}{\sigma_i} v_{\sigma i}. \tag{1.4}$$

is known as a *Truncated SVD solution* (TSVD). Alternatively, we could use *Tikhonov regularization.* As shown by Varah [22], these two techniques will often give similar results. In general, the ability to acquire information about the largest singular values and vectors of A is essential in solving discrete ill-posed problems. We refer the reader to [12] for an excellent review of regularization techniques for ill-posed problem.

For large ill-posed problems it is not usually feasible to compute the SVD (1.2). Therefore iterative methods which only use the cheap operations Av and $A^T u$ are attractive alternatives, see [11] and [12]. Regularization can now be achieved by terminating the iteration after $k = k(\epsilon) \ll n$ steps. In particular, step k of the conjugate gradient method applied to the normal equations, CGLS (or CGNR), solves

$$\min_{x_k \in \mathcal{K}_k} \|Ax_k - b\|_2, \qquad k = 1, 2, \ldots, \tag{1.5}$$

where $\mathcal{K}_k$ is the Krylov subspace of dimension k

$$\mathcal{K}_k = span\{A^T b, (A^T A)A^T b, \ldots, (A^T A)^{k-1} A^T b\}. \tag{1.6}$$

For a more detailed description of CGLS see [2, Sec. 20].

CGLS often converges much faster than other iterative methods for such problems, see also [15] and [3]. However, it has been observed, see [11], that the conjugate gradient method also *diverges* much more rapidly than competing iterative methods. It is therefore essential to stop iterating after an optimal number of steps k_*, before the effects of the smaller singular values begin to appear. Unfortunately, it is possible for some of these smaller singular values to appear before all of the larger singular values are well approximated, see [12].

In this paper, we study methods based on the Lanczos bidiagonalization (LBD) used in LSQR, see [18]. LSQR computes in exact arithmetic the same sequence of approximate solutions as CGLS. In [16] and [1] methods are developed which combine LBD with regularization in the Krylov space such as Tikhonov regularization or truncated SVD (TSVD) to yield *hybrid* methods. Ideally, this combined approach will damp the effect of small singular values on x_k, so that the only disadvantage of overestimating k_* in a hybrid iterative method is the extra work involved in performing the unneeded iterations.

We begin in §2 by reviewing the Lanczos bidiagonalization method for ill-posed problems and by surveying methods from [1] for regularizing in the Krylov space. In §3 we discuss the use of generalized cross-validation to estimate the optimal regularization of an LBD solution when the noise level is unknown. For this approach to be applicable, it is essential that LBD is run with complete reorthogonalization, so the technique does not apply to CGLS. We give an example which shows that many steps may be needed to get a reliable determination of the optimal regularization parameter, ν_*. This fact motivates the study in §4 of the implicitly restarted LBD method, based on techniques developed in [19]. We explore the use of implicit restarts in the context of determining regularization parameters for the LBD in §5. Numerical examples are presented to demonstrate that a hybrid iterative method which incorporates implicit restarts can typically obtain the larger singular values of A and an accurate regularization parameter when k is only slightly larger than optimal. Comparisons of the efficiency of this restarted hybrid method versus existing techniques are also provided.

2 Lanczos bidiagonalization for ill-posed problems

A Lanczos algorithm for the bidiagonalization of a rectangular matrix A was first given by Golub and Kahan [8]. Paige and Saunders [18] describe two different variants: Bidiag 1 starts with the m-vector b and gives a reduction to *lower* bidiagonal form; Bidiag 2 starts with the n vector A^Tb and reduces A to *upper* bidiagonal form. (Bidiag 2 is the procedure originally given by Golub and Kahan.) These two Lanczos bidiagonalization algorithms are closely related to Lanczos method applied to the real symmetric matrices AA^T and A^TA, or alternatively to $\begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix}$.

2.1 Lanczos bidiagonalization

Paige and Saunders [18] showed that Bidiag 1 is the more stable algorithm for solving least squares problems, and the following description of this algorithm is adopted from their paper. We assume here that *exact* arithmetic is employed and discuss the effects of limited precision later.

Let the starting vectors $u_1 \in R^m$ and $v_1 \in R^n$ be defined by

$$\beta_1 u_1 = b, \qquad \alpha_1 v_1 = A^T u_1, \tag{2.1}$$

and for $i = 1, 2, \ldots$ compute

$$\beta_{i+1} u_{i+1} = A v_i - \alpha_i u_i, \tag{2.2}$$

$$\alpha_{i+1} v_{i+1} = A^T u_{i+1} - \beta_{i+1} v_i, \tag{2.3}$$

where $\beta_{i+1} \geq 0$ and $\alpha_{i+1} \geq 0$ are normalization constants chosen so that $\|u_{i+1}\|_2 = \|v_{i+1}\|_2 = 1$. In particular, $\beta_1 = \|b\|_2$.

With the definitions

$$U_k = (u_1, u_2, \ldots, u_k), \qquad V_k = (v_1, v_2, \ldots, v_k),$$

and

$$B_k = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix} \in R^{(k+1)\times k}, \tag{2.4}$$

the recurrence relations (2.2) and (2.3) can be written as

$$A^T U_{k+1} = V_k B_k^T + \alpha_{k+1} v_{k+1} e_{k+1}^T, \tag{2.5}$$

$$AV_k = U_{k+1} B_k. \tag{2.6}$$

Note that here and in the following e_j denotes the j^{th} column of a unit matrix of appropriate dimension.

Seeking an approximate solution $x_k \in \mathcal{K}_k = span(V_k)$, we write

$$x_k = V_k f_k.$$

Multiplying equation (2.6) by f_k, we obtain $Ax_k = AV_k f_k = U_{k+1} B_k f_k$, and since $b = \beta_1 U_{k+1} e_1$ we have

$$b - Ax_k = U_{k+1}(\beta_1 e_1 - B_k f_k). \tag{2.7}$$

From the orthogonality of U_{k+1}, it follows that $\|b - Ax_k\|_2$ is minimized over all $x_k \in \mathcal{R}(V_k)$ by taking f_k to be the solution to the least squares problem

$$\min_{f_k} \|\beta_1 e_1 - B_k f_k\|_2. \tag{2.8}$$

Assume that the bidiagonalization can be carried out for $i = 1, \ldots, k-1$ without breakdown. The recurrence (2.2)–(2.3) terminates for $i = k$ if either $\beta_{k+1} = 0$ or $\alpha_{k+1} = 0$. If $\beta_{k+1} = 0$, it follows that

$$(AA^T)^k b \in span\{b, (AA^T)b, \ldots, (AA^T)^{k-1}b\}.$$

In this case, (2.7) holds with $\beta_{k+1} = 0$ and u_{k+1} an arbitrary vector in the orthogonal complement to $\mathcal{R}(U_k)$. Since $\alpha_1, \ldots, \alpha_k \neq 0$ it follows that we can determine f_k so that $b - AV_k f_k = 0$, which implies $b \in \mathcal{R}(A)$. Hence this case can only occur if the linear system $Ax = b$ is consistent. If $\beta_{k+1} \neq 0$, but $\alpha_{k+1} = 0$, then using (2.5) we have

$$A^T(b - AV_k f_k) = A^T U_{k+1}(\beta_1 e_1 - B_k f_k) = V_k B_k^T(\beta_1 e_1 - B_k f_k) = 0.$$

Here the last equality follows since f_k satisfies the normal equations. Thus $x_k = V_k f_k$ solves the least squares problem (1.1) exactly for some finite k corresponding to a termination of the Lanczos recurrences.

We now consider the use of a more general starting vector $u_1 \in R^m$, $\|u_1\|_2 = 1$. Assume that the bidiagonalization has been carried out for $i = 1, \ldots, k$ without breakdown. To minimize $\|b - Ax_k\|_2$ over all $x_k \in \mathcal{R}(V_k)$ we should take $x_k = V_k f_k$ and solve the subproblem

$$\min_{f_k} \|b - U_{k+1} B_k f_k\|_2. \tag{2.9}$$

When $u_1 \neq b/\|b\|_2$, then in general $b \notin \mathcal{R}(U_{k+1})$, and we let

$$b = b_1 + b_2, \qquad b_1 = \beta_1 U_{k+1} c, \qquad b_2 \perp b_1, \tag{2.10}$$

where b_1 is the orthogonal projection of b onto $\mathcal{R}(U_{k+1})$. Then $b - Ax_k = U_{k+1}(\beta_1 c - B_k f_k) + b_2$, and in analogy to (2.8) we take f_k as the solution to

$$\min_{f_k} \|\beta_1 c - B_k f_k\|_2. \tag{2.11}$$

In Section 4 we will consider implicitly generating the bidiagonalization corresponding to a starting vector

$$\hat{\beta}_1 \hat{u}_1 = \Psi_p(AA^T)b = (AA^T - \mu_p I)(AA^T - \mu_{p-1}) \cdots (AA^T - \mu_1 I)b, \tag{2.12}$$

which can be interpreted as a polynomial filter in AA^T applied to b. Note that when $\hat{u}_1$ is the starting vector, we will not in general be able to determine an exact solution to (1.1) in a finite number of iterations even if exact arithmetic is utilized. However, starting vectors satisfying (2.12) will be useful for determining better approximations to truncated singular value solutions. It should be stressed that in practice we are interested in obtaining an approximate solution with significantly fewer iterations than typically required for terminating the recurrence.

2.2 Reorthogonalization in Lanczos bidiagonalization

The Lanczos bidiagonalization will yield approximations to the extreme singular values of A. In finite precision it is well known (see [17] for an analysis of the symmetric case) that the convergence of the singular values of B_k is accompanied with a loss of orthogonality in U_k and V_k. In general, the relations (2.5) and (2.6) still hold to full precision, and the loss of orthogonality will not affect the final accuracy of the approximations.

Lanczos methods using reorthogonalization are discussed in [10, Sec. 9.2]. In LDB, the loss of orthogonality can be avoided by reorthogonalizing the newly computed vectors u_{i+1} and v_{i+1} against all of their predecessors. Although this method, denoted LBDR, is much more costly in terms of storage and operations, the extra cost may be acceptable as long as the number of steps k is small. A great advantage is that LBDR allows the use of stopping rules which do not depend on a priori knowledge of the size of noise on b, see §3.

2.3 Regularization in the Krylov space

In developing the Lanczos method for computing x_k, it was assumed that we wanted to approximate the exact solution to (1.1). However, as motivated in §1, an approximation to a *regularized* solution is desired for the ill-posed problem. To achieve this approximation, we now generalize the approach in [1].

Let the singular value decomposition of the matrix $B_k \in R^{(k+1)\times k}$ in (2.4) be

$$B_k = P_{k+1} \begin{pmatrix} \Omega_k \\ 0 \end{pmatrix} Q_k^T = \sum_{i=1}^{k} \omega_i p_i q_i^T, \tag{2.13}$$

where $P_{k+1} \in R^{(k+1)\times(k+1)}$ and $Q_k \in R^{k\times k}$ are square orthogonal matrices and $\omega_1 \geq \omega_2 \geq \cdots \geq \omega_k > 0$. Then the solution to (2.11) can be written

$$f_k = \beta_1 Q_k \begin{pmatrix} \Omega_k^{-1} & 0 \end{pmatrix} P_{k+1}^T c = \beta_1 \sum_{i=1}^{k} \frac{\gamma_i}{\omega_i} q_i, \qquad \gamma = P_{k+1}^T c. \tag{2.14}$$

Regularized approximations

$$x_{k,\nu} = V_k f_{k,\nu} \quad \text{where} \quad f_{k,\nu} = \beta_1 \sum_{i=1}^{\nu} \frac{\gamma_i}{\omega_i} q_i, \qquad \nu \leq k, \tag{2.15}$$

can now be obtained for $k = 1, 2, \ldots$ by truncating the expansion (2.14). Using the orthogonality of V_k and Q_k,

$$\|x_{k,\nu}\|_2^2 = \|V_k f_{k,\nu}\|_2^2 = \beta_1^2 \sum_{i=1}^{\nu} (\gamma_i/\omega_i)^2. \tag{2.16}$$

Combining (2.13) and (2.15),

$$B_k f_{k,\nu} = (\sum_{i=1}^{k} \omega_i p_i q_i^T) \beta_1 \sum_{j=1}^{\nu} \frac{\gamma_j}{\omega_j} q_j = \beta_1 \sum_{i=1}^{\nu} \gamma_i p_i,$$

and since $\beta_1 c = \beta_1 P_{k+1} P_{k+1}^T c = \beta_1 P_{k+1} \gamma$, the corresponding residual vector is

$$d_{k,\nu} = \beta_1 c - B_k f_{k,\nu} = \beta_1 \sum_{i=\nu+1}^{k+1} \gamma_i p_i. \tag{2.17}$$

From (2.10) $r_{k,\nu} = b - A x_{k,\nu} = b_2 + U_{k+1} d_{k,\nu}$, and since $u_2 \perp \mathcal{R}(U_{k+1})$ we obtain

$$\|r_{k,\nu}\|_2^2 = \|b_2\|_2^2 + \beta_1^2 \sum_{i=\nu+1}^{k+1} \gamma_i^2. \tag{2.18}$$

Finally, for the residual to the normal equation using (2.5) (cf. [1])

$$\begin{aligned} A^T r_{k,\nu} &= A^T b_2 + A^T U_{k+1} d_{k,\nu} \\ &= A^T b_2 + \beta_1 V_k \sum_{i=\nu+1}^{k} \omega_i \gamma_i p_i + \beta_1 \alpha_{k+1} \sum_{i=\nu+1}^{k+1} \gamma_i e_{k+1} p_i \end{aligned} \tag{2.19}$$

and hence

$$\|A^T r_{k,\nu}\|_2^2 = \|A^T b_2\|_2^2 + \beta_1^2 \sum_{i=\nu+1}^{k} \gamma_{1i}^2 \omega_i^2 + (\beta_1 \alpha_{k+1})^2 \sum_{i=\nu+1}^{k+1} \gamma_i^2 \kappa_{k+1,i}^2 \tag{2.20}$$

where $\kappa_{ji} = (P_{k+1})_{ji}$.

Note that the norms of $x_{k,\nu}$, $r_{k,\nu}$ and $A^T r_{k,\nu}$ can be computed without explicitly forming $x_{k,\nu}$ or $f_{k,\nu}$. We only need the singular values of B_k and the vectors $P_{k+1}^T c$ and $P_{k+1}^T e_{k+1}$. For the significant case when $\beta_1 u_1 = b$, we have $c = e_1$, $\gamma_i = \kappa_{1i}$, and in the formulas above only the elements in the *first and last rows* of P_{k+1} are required.

Although we have here only considered regularization by *truncating* the singular value expansion, similar techniques can also be developed for Tikhonov regularization.

3 Generalized Cross Validation

When the error norm, $||\epsilon r||$, is known with good accuracy, the discrepancy principle can be used as a stopping rule for CGLS and LBD. The iterations are stopped when the residual norm $||b - Ax_k||_2$ is equal to a predetermined upper bound for the noise in the right hand side. However, under-estimation of the noise level may cause failure, see the discussions in [13] and [12]. We consider here the use of Generalized Cross Validation (GCV), due to Golub, Heath and Whaba [7], as a stopping rule. This stopping criterion has the important advantage that no a priori information about the noise level is required.

In GCV the regularization parameter is chosen as follows. Let ν be the regularization parameter and $A^\dagger(\nu)$ be the matrix which maps the right hand side onto the regularized solution x_ν. Then our estimate for ν_* (the value of ν such that $||x_\nu - \bar{x}||_2$ is minimized) is chosen as the minimizer of the function (see [7])

$$G(\nu) = \frac{\frac{1}{m}||(I - AA^\dagger(\nu))b||_2^2}{\left(\frac{1}{m}trace(I - AA^\dagger(\nu))\right)^2}, \qquad \nu = 1 \ldots n. \tag{3.1}$$

The GCV estimate of ν_* is known to have a number of favorable properties both from practical experience and theory. However, the theory is an asymptotic one and good results cannot be expected when there is not enough information in the data to separate signal from noise. For a review of results on GCV, see [21, Ch.4].

3.1 Generalized cross validation for TSVD

Let the singular value decomposition be given by (1.2), and consider the truncated SVD solution (1.4). An elementary calculation gives $trace(I - AA^\dagger(\nu)) = m - \nu$, and

$$||(I - AA^\dagger(\nu))b||_2^2 = ||b - Ax_\nu||_2^2 = \sum_{i=\nu+1}^{m} (u_{\sigma i}^T b)^2, \tag{3.2}$$

where $u_{\sigma i}$ is the i^{th} left singular vector of A. The residual norm provides a measure of the last $m - \nu$ rhs coefficients of the problem. In fact, if ν can be selected such that we have $u_{\sigma i}^T b \approx \epsilon u_{\sigma i}^T r$ if and only if $i > \nu$, then (3.2) provides in a sense the best possible estimate of the noise on b. For this case the GCV function can be interpreted as

$$\begin{aligned} G(\nu) &= \left(\frac{m}{(m-\nu)}\right) \cdot \left(\frac{\sum_{i=\nu+1}^{m}(u_{\sigma i}^T b)^2}{m-\nu}\right) \\ &= \text{(Weighting)} \cdot \text{(Variance Estimate)}, \end{aligned} \tag{3.3}$$

see also [13]. The summation on the right side of (3.3) yields a variance estimate of $||r\epsilon||$ which is based on the last $m - \nu$ rhs coefficients. When ν is too small, desirable rhs coefficients, $u_{\sigma i}^T b > \epsilon u_{\sigma i}^T r$, enter into the variance estimate and the cross-validation function is not minimized for this ν. On the other hand, when ν is too large, the weighting term $m/(m - \nu)$ in (3.3) grows large so that the cross-validation function is not minimized at this value of ν.

In the following, (3.3) will be denoted the *full-information* GCV function to signify that all of the singular values and vectors of A are available in the expression.

3.2 Generalized cross validation for LBDR

As pointed out in [12], assuming $\nu_* = k$ and using GCV as a stopping rule does not seem to be feasible for CGLS because, due to the loss of orthogonality, no simple way exists to determine the denominator in (3.1). Another problem is that CGLS depends on the right hand side b in a complicated nonlinear way. To overcome these problems Girard [6] has suggested the Monte-Carlo GCV.

We now consider the use of cross validation for the LBDR algorithm with regularization. It should be noted that we consider here only a linear model, and neglect the nonlinear behavior of the LBDR approximations as function of b. To evaluate the GCV function for the set of regularized solutions $x_{k,\nu}$ in (2.15), we first note that $r_{k,\nu} = (I - AA^\dagger(k,\nu))b$, and hence by (2.18)

$$\|(I - AA^\dagger(\nu))b\|_2^2 = \|b_2\|_2^2 + \beta_1^2 \sum_{i=\nu+1}^{k+1} \gamma_i^2. \tag{3.4}$$

For LBD with full reorthogonalization, a simple expression for the denominator of the GCV function can be derived. Provided that U_{k+1} is orthogonal we have from (2.10) $\beta_1 c = U_{k+1}^T b$, and hence

$$x_{k,\nu} = A^\dagger(k,\nu)b = V_k B_{k,\nu}^\dagger U_{k+1}^T b, \quad B_{k,\nu}^\dagger = Q_k \begin{pmatrix} \Omega_{k,\nu}^{-1} & 0 \\ 0 & 0 \end{pmatrix} P_{k+1}^T. \tag{3.5}$$

Using (2.13) it follows that

$$AA^\dagger(k,\nu) = AV_k B_{k,\nu}^\dagger U_{k+1}^T = U_{k+1} B_k B_{k,\nu}^\dagger U_{k+1}^T, \tag{3.6}$$

where

$$B_k B_{k,\nu}^\dagger = P_{k+1} \begin{pmatrix} \Omega_k \\ 0 \end{pmatrix} Q_k^T Q_k \begin{pmatrix} \Omega_{k,\nu}^{-1} & 0 \\ 0 & 0 \end{pmatrix} P_{k+1}^T.$$

By the orthogonality of Q_k we have

$$AA^\dagger(k,\nu) = U_{k,\nu} U_{k,\nu}^T, \qquad U_{k,\nu} = U_{k+1} P_{k+1} \begin{pmatrix} I_\nu \\ 0 \end{pmatrix} \in R^{m\times\nu},$$

where $U_{k,\nu}$ has orthogonal columns, and hence

$$trace(I_m - AA^\dagger(k,\nu)) = m - trace(U_{k,\nu}^T U_{k,\nu}) = m - \nu. \tag{3.7}$$

Note that the orthogonality of V_k was not used in this derivation.

Using (3.4) and (3.7) in the GCV function, it follows that the number of terms, $\nu_* \le k$, to include should be estimated as the minimizer of

$$G(k,\nu) = \frac{m}{(m-\nu)^2}\Big(\|b_2\|_2^2 + \beta_1^2 \sum_{i=\nu+1}^{k+1} \gamma_i^2\Big), \qquad \nu = 1,\dots,k. \tag{3.8}$$

For the special case when $\beta_1 u_1 = b$ we have $b_2 = 0$, and $\gamma_i = \kappa_{1i}$. Hence in this case all the information needed to compute (3.8) is contained in the first row of the matrix P_k.

We remark that from

$$b_1 = \beta_1 U_{k+1} P_{k+1} P_{k+1}^T c = \beta_1 U_{k+1} P_{k+1} \gamma,$$

it follows that the norm of the residual using LBD with full reorthogonalization can be expressed as

$$\|r_{k,\nu}\|_2^2 = \|b\|_2^2 + \beta_1^2 \sum_{i=\nu+1}^{k+1} \gamma_i^2 - \|b_1\|_2^2 = \|b\|_2^2 - \beta_1^2 \sum_{k=1}^{\nu} \gamma_i^2.$$

Because $\beta_1 c = U_{k+1}^T b$, this expression for the residual norm leads to a way of rewriting the GCV function,

$$G(k,\nu) = \frac{m}{(m-\nu)^2}\Big(\|b\|_2^2 - \sum_{i=1}^{\nu}(p_i^T U_{k+1}^T b)^2\Big), \qquad \nu = 1,\ldots,k, \tag{3.9}$$

which depends only on the rhs b and the vectors $U_{k+1}p_i$, $i \le \nu$.

3.3 Application of GCV to an ill-posed problem

To illustrate the use of GCV with LBD, we consider the least squares test problem deriv2 from the MATLAB package of regularization tools by Hansen [14]. This problem is mildly ill-posed and arises from the discretization by collocation with piecewise linear splines of the integral equation

$$\int_0^1 k(s,t)x(t)dt = g(s), \qquad s \in [0,1], \tag{3.10}$$

where the kernel is Green's function for the second derivative

$$k(s,t) = \begin{cases} s(t-1), & \text{if } 0 \le s \le t \le 1; \\ t(s-1), & \text{if } 0 \le t < s \le 1. \end{cases} \tag{3.11}$$

The right hand side $g(s)$ and the corresponding solution are

$$g(s) = \exp(s) + (1-e)s - 1, \qquad f(t) = \exp(t).$$

We take $m = n = 50$, and add normally distributed noise $\mathcal{N}(0, 10^{-p})$, $p = 4$.

Unless stated otherwise, the following examples were performed in MATLAB using double precision, and with the seed for the random generator set to zero. Regularization and analysis routines from [14] were used, sometimes in slightly modified form. We first solved this problem by computing the SVD of the full matrix $A \in R^{50\times 50}$. The minimum of the GCV function occurs for $\nu = 9$, as shown in Figure 3.1. Note that this minimum varies with the seed for the random noise.

To investigate how LBDR performs, we also plot the GCV function corresponding to the Lanczos factorization of size 20 in Figure 3.1. Although the first points of $G(20,\nu)$ match the full-information GCV curve closely, the GCV curve for LBDR drops off sharply as ν approaches k. This behavior is due to the fact that LBDR yields accurate approximations to the larger singular values and corresponding singular vectors of A [9] (assuming that the singular values are unique for the time being). For values of ν less than and away from k, $u_{\sigma\nu} \approx U_{k+1}p_\nu$ so that comparing (3.2) and (3.9) yields $G(\nu) \approx G(20,\nu)$. On the other hand, when $\nu \approx k$, $U_{k+1}p_\nu$ and $G(20,\nu)$ can deviate significantly from $u_{\sigma\nu}$ and $G(\nu)$ respectively. A sharp drop off of the LBDR GCV curve as ν nears k appears to be typical.

The global minimum at $\nu = 9$ on the full-information GCV curve is well represented in $G(20,\nu)$. However, because of the undesirable drop off past $\nu = 19$, this minimum is only a local one. As a result, a naive search for the global minimum of $G(k,\nu)$ leads to difficulties. The GCV curve of LBDR cannot generally be trusted for values of ν near k.

A simple approach to remedying this problem is to "weigh" the GCV function so that a minimum cannot occur at a ν near k. For instance, assign the optimal regularization parameter to be the minimizer of $G(k,\nu)$ in (3.8) for $\nu = 1,\ldots,(k-\delta)$, which corresponds to applying an "infinite weighting" at those values of ν greater than $(k-\delta)$. When δ is properly chosen, a legitimate minimum of the GCV function is chosen as the estimate for the desired regularization parameter, ν_*. For example, Figure 3.2 displays the minimizer of $G(k,\nu)$ as k varies and for $\delta = 1, 3$ and 5. As Figure 3.1 shows, $\delta = 1$ is not always sufficient for damping out the false minimums at $\nu \approx k$. For $\delta = 3$, the corresponding plot has converged to 9 already at $k = 12$. The value $\delta = 5$ also yields the regularization parameter 9, although not until $k = 14$.

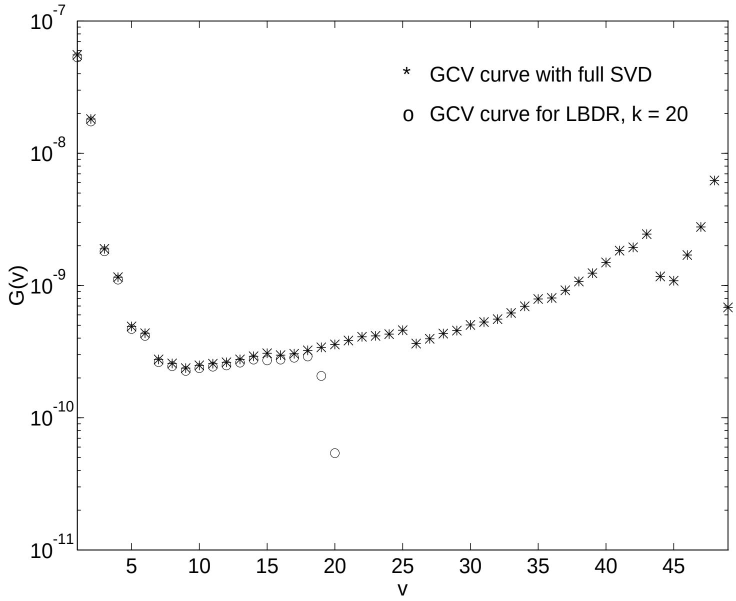

Figure 3.1: Cross validation curves for the problem deriv2 when (i) the exact SVD and (ii) an LBDR factorization of size $k = 20$ are known.

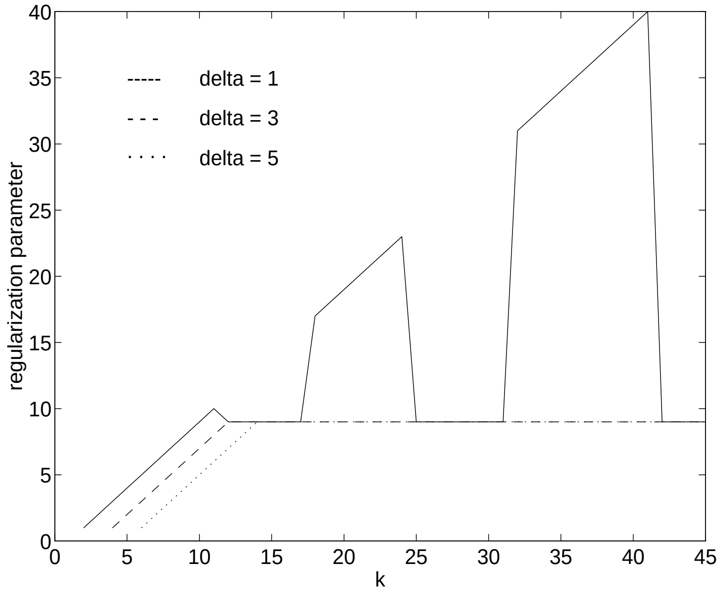

Figure 3.2: Estimate of the optimal regularization parameter (i.e., the value of $\nu \leq (k - \delta)$ which is a minimizer of the weighted GCV function) as k and δ vary.

When modified to insure that only converged SVD data is involved in estimating ν_*, GCV typically yields an accurate result in an efficient and reliable manner. However, weighting the GCV function is a heuristic technique which depends on both the problem being solved and on the level of noise on b. If the weighting is too small (non-converged data is considered by GCV), false minima will occur in the GCV function. If too much weighting is applied, converged data is not considered by GCV, and k must grow large before the correct regularization parameter is found.

Running the LBDR algorithm for a large number of steps k becomes very expensive because of the reorthogonalization. Thus an adaptive technique is desired which insures that the computation of the regularization parameter is determined solely by converged SVD data. This goal is one of the motivations for studying the implicitly restarted LBDR algorithm.

4 The implicitly restarted bidiagonalization algorithm

The bidiagonal decomposition generated by a Lanczos approach is uniquely determined by the matrix A and a user-specified starting vector, $u_1 \in R^m$. Given that a decomposition corresponding to the pair $\{A, u_1\}$ already exists, an implicit technique is developed in §4.1 which generates a new LBD decomposition corresponding to the starting vector $(AA^T - \mu_1 I)u_1$. This approach can be repeated, and after p restarts it corresponds to applying a polynomial filter, $\Psi_p(AA^T)$, to u_1 as in (2.12).

The impetus for restarting LBDR is the implicitly restarted Arnoldi method [19] in which implicit restarts are used to accurately compute a portion of the spectrum of a square matrix A. Analogous strategies may be invoked in conjunction with a restarted bidiagonalization algorithm in order to determine a *fixed* number of the largest singular values and corresponding vectors of A. In particular, the "exact" shift strategies and theorems of [19] can be adapted to the singular value problem in a straightforward fashion.

Since we want to damp small singular values, the simplest procedure is to choose zero shifts $\mu = 0$. Then p shifts correspond to starting the LBDR with the vector

$$\hat{\beta}_1 \hat{u}_1 = (AA^T)^p b. \tag{4.1}$$

Besides being simple to implement, zero shifts are in fact extremely effective for the regularization problem because the number of singular values to be retained is not known a priori. Note that using the starting vector in (3.1) simply corresponds to working with a subspace derived by *deleting* the first p vectors spanning the Krylov subspace of dimension $p + k$

$$\mathcal{K}_{p+k} = span\{A^T b, (A^T A)A^T b, \ldots, (A^T A)^{p+k-1} A^T b\}.$$

We stress that the implicitly restarted LBDR method holds several significant advantages over explicitly restarting with the new starting vector. In particular, the implicit shift algorithm generates the corresponding LBDR decomposition in a stable way. LBDR, on the other hand, will perform poorly when explicitly started with the vector, $\hat{u}_1$, defined in (4.1). For large values of p, one is essentially starting with a good approximation to the singular vector corresponding to the largest singular value of A. Additionally, it will be shown that an implicit restart is more efficient than its explicit counterpart. We will return to these issues in greater detail in §4.2 and §4.3.

4.1 Implicit bidiagonalization restarts

Combining (2.5) and (2.6) yields traditional symmetric Lanczos relations,

$$(A^T A)V_k = V_k(B_k^T B_k) + \alpha_{k+1}\beta_{k+1} v_{k+1} e_{k+1}^T, \tag{4.2}$$

$$(AA^T)U_{k+1} = U_{k+1}(B_k B_k^T) + \alpha_{k+1} A v_{k+1} e_{k+1}^T. \tag{4.3}$$

At this point, one might consider working with the products $M = AA^T$ and $T_k = B_k B_k^T$ and applying existing implicit restart techniques [19, 4] to (4.3) in order to obtain the decomposition corresponding to $\hat{u}_1$. However, to preserve the sparsity and degree of conditioning of the original problem, we would much rather work directly with (2.5) to find the restarted recurrence relations. As a consequence, and contrary to previous implicit restart strategies, the following method will employ SVD steps on B_k rather than QR steps on $B_k^T B_k$.

The first step in the implicit bidiagonalization restart is a Golub-Kahan SVD step with shift μ, see [10, Alg. 8.3.1]. Following the SVD algorithm, first find a Givens rotation $G^{(1)} = G_{1,2}(\theta_1)$ such that

$$\begin{bmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{bmatrix} \begin{bmatrix} \alpha_1^2 - \mu \\ \alpha_1\beta_2 \end{bmatrix} = \begin{bmatrix} * \\ 0 \end{bmatrix}.$$

One could then compute additional Givens rotations such that $Q^T B_k B_k^T Q$ is tridiagonal where $Q^T = G^{(k)} \cdots G^{(2)} G^{(1)}$, i.e., perform an implicit QR iteration on $B_k B_k^T$. Yet for the reasons given above, this action is to be avoided. Rather, one computes a series of Givens rotations $G_l^{(i)} \in R^{(k+1),(k+1)}$ and $G_r^{(i)} \in R^{k,k}$ such that

$$\hat{B}_k = G_l^{(k)} \cdots G_l^{(2)} G^{(1)} B_k G_r^{(1)} \cdots G_r^{(k-1)} \tag{4.4}$$

is bidiagonal. Note that the rotations $G_l^{(i)} = G_{i,i+1}(\theta_i)$ and $G_r^{(i)} = G_{i,i+1}(\theta_i)$ are used to chase the unwanted nonzero term of $G^{(1)}B_k$ down the bidiagonal. Define $Q_l^T = G_l^{(k)} \cdots G_l^{(2)} G_1$ and $Q_r = G_r^{(1)} \cdots G_r^{(k-1)}$. Then by the implicit Q theorem [10], Q and Q_l are related by

$$Q_l^T B_k B_k^T Q_l = D^T Q^T B_k B_k^T Q D,$$

so that

$$Q_l = QD, \qquad D = \mathrm{diag}(1, \pm 1, \ldots, \pm 1). \tag{4.5}$$

It should also be noted that Q_l and Q_r are both orthogonal matrices, being products of Givens rotations.

Now define $\hat{U}_{k+1} = U_{k+1} Q_l$ and $\hat{V}_k = V_k Q_r$, while recalling that $\hat{B}_k = Q_l^T B_k Q_r$. Multiplying (2.5) on the right by Q_l and including an identity factor $I = Q_r Q_r^T$ yields

$$A^T (U_{k+1} Q_l) = (V_k Q_r) Q_r^T B_k^T Q_l + \alpha_{k+1} v_{k+1} e_{k+1}^T Q_l.$$

Using our previous definitions, this expression can be rewritten to yield the new recursive relation

$$A^T \hat{U}_{k+1} = \hat{V}_k \hat{B}_k^T + \alpha_{k+1} v_{k+1} e_{k+1}^T Q_l. \tag{4.6}$$

Unfortunately, (4.6) is not a valid recursive relation for the bidiagonalization algorithm. The residual term of (4.6) takes on an invalid form

$$\text{residual} = \alpha_{k+1} v_{k+1} \begin{bmatrix} 0 & \ldots & 0 & q_{l_{(k+1,k)}} & q_{l_{(k+1,k+1)}} \end{bmatrix} \tag{4.7}$$

where $q_{l_{(i,j)}}$ is the (i,j) element of Q_l. This difficulty is consistent with those encountered in the development of the implicitly restarted Arnoldi algorithm [19]. To deal with this problem we need to "back up" one step, and first rewrite (4.6) as

$$A^T (\hat{U}_k, \hat{u}_{k+1}) = (\hat{V}_{k-1}, \hat{v}_k, v_{k+1}) \begin{pmatrix} \hat{B}_{k-1}^T & 0 \\ \hat{\alpha}_k e_k^T & \hat{\beta}_{k+1} \\ \alpha_{k+1} q_{l_{(k+1,k)}} e_k^T & \alpha_{k+1} q_{l_{(k+1,k+1)}} \end{pmatrix}. \tag{4.8}$$

Then equating the first k columns of (4.8), one obtains

$$A^T \hat{U}_k = \hat{V}_{k-1} \hat{B}_{k-1}^T + \hat{r}_k e_k^T \tag{4.9}$$

where the new residual is

$$\hat{r}_k = \hat{\alpha}_k \hat{v}_k + \alpha_{k+1} q_{l_{(k+1,k)}} v_{k+1}.$$

The recurrence relation of (4.9) is a valid relation for the bidiagonalization algorithm and is the restarted analogue to (2.5). In obtaining this new relation, one step of the algorithm was sacrificed.

It is a simple matter to show that $\hat{U}_{k+1}^T \hat{U}_{k+1} = I$, $\hat{V}_k^T \hat{V}_k = I$, and $A\hat{V}_k = \hat{U}_{k+1}\hat{B}_k$. The restarted analogue of (2.6) can then be obtained as

$$A\hat{V}_{k-1} = \hat{U}_{k+1} \begin{pmatrix} \hat{B}_{k-1} \\ 0 \end{pmatrix} = \hat{U}_k \hat{B}_{k-1}.$$

For completeness, it should also be noted that

$$\hat{V}_{k-1}^T \hat{r}_k = \hat{\alpha}_k \hat{V}_{k-1}^T \hat{v}_k + \alpha_{k+1} Q_{r_{k-1}}^T V_k^T v_{k+1} = 0$$

where $Q_{r_{k-1}}$ contains the first $k-1$ columns of Q_r.

A new recurrence relation was found; but nothing has been shown concerning $\hat{u}_1$ and its relationship to u_1. To see this relationship, recall (4.3) and subtract μI from both sides to obtain

$$(AA^T - \mu I)U_{k+1} = U_{k+1}(B_k B_k^T - \mu I) + \alpha_{k+1} A v_{k+1} e_{k+1}^T. \tag{4.10}$$

From the first stage of the QR step on $B_k^T B_k$ with shift μ, we have

$$Q^T (B_k B_k^T - \mu I) = R,$$

where R is upper triangular. Using (4.5), and multiplying (4.10) on the right by e_1 yields

$$(AA^T - \mu I)u_1 = (U_{k+1} Q_l D) R e_1 = \pm (U_{k+1} Q_l) e_1 r_{1,1}.$$

Thus the new starting vector does indeed satisfy the condition,

$$\hat{u}_1 = \pm r_{1,1}^{-1} (AA^T - \mu I) u_1,$$

assuming α_1 (and thus $r_{1,1}$) is not equal to zero. Furthermore, one has knowledge of $\hat{B}_{k-1}$, $\hat{U}_k$, and $\hat{V}_{k-1}$ without explicitly restarting the entire bidiagonalization algorithm.

For the right hand side b we assume that before the restart

$$b = b_1 + b_2, \qquad b_1 = \beta_1 U_{k+1} c, \qquad b_2 \perp \mathcal{R}(U_{k+1}).$$

Hence after a single restart

$$b_1 = \beta_1 \hat{U}_{k+1} Q_l^T c = \beta_1 \hat{U}_{k+1} \begin{pmatrix} \hat{c} \\ \gamma/\beta_1 \end{pmatrix} = \beta_1 \hat{U}_k \hat{c} + \gamma \hat{u}_{k+1},$$

and hence

$$b = \hat{b}_1 + \hat{b}_2, \qquad \hat{b}_1 = \beta_1 \hat{U}_k \hat{c}, \qquad \hat{b}_2 = b_2 + \gamma \hat{u}_{k+1}. \tag{4.11}$$

To minimize the residual $b - A\hat{x}_{k-1}$ where $\hat{x}_{k-1} = \hat{V}_{k-1}\hat{f}_{k-1}$, we choose $\hat{f}_{k-1}$ to be the solution to the least squares problem

$$\|\beta_1 \hat{c} - \hat{B}_{k-1}\hat{f}_{k-1}\|_2.$$

The above derivation only handles a single shift, μ. But as the results of a single shift still define a valid bidiagonal decomposition, it should be clear that one can repeatedly perform shifts until the desired polynomial filter is applied to u_1. In this case, a series of transformations

$$\hat{Q}_l = Q_l^{(p)} \cdots Q_l^{(1)} \quad \text{and} \quad \hat{Q}_r = Q_r^{(1)} \cdots Q_r^{(p)}$$

would be applied to B_k, U_{k+1} and V_k. A new decomposition of size $k-p$ results where $\hat{B}_{k-p}$, $\hat{U}_{k-p}$, and $\hat{V}_{k-p}$ are the appropriate leading portions of $\hat{Q}_l^T B_k \hat{Q}_r$, $U_{k+1}\hat{Q}_l$ and $V_k\hat{Q}_r$ respectively. Most importantly, the first column of $U_{k+1}\hat{Q}_l$ satisfies (2.12).

4.2 Cost analysis of implicit LBDR restarts

Table 4.1 summarizes the dominating computational costs involved in a single implicit LBDR restart, assuming that $k \ll n$. Each LBDR iteration and each restart requires one product Av and $A^T v$, which could dominate the work requirement in some applications. In Table 4.1 it is assumed that each of these products take αm operations, where the parameter α represents the average number of nonzero entries in a row of A.

Approximately $8k(m+n) + 2\alpha m$ operations are required in the restart. Computing $\hat{U}_{k+1} = U_{k+1}Q_l$ and $\hat{V}_k = V_k Q_r$ require about $6mk$ and $6nk$ operations respectively; the remaining work is necessary to recover the lost LBDR iteration. Note that these costs compare favorably to the explicit LBDR restart which requires $O(k^2 m)$ operations. With implicit restarts, one obtains a fully orthogonalized factorization with the same order of computations required by an explicitly restarted *non-reorthogonalized* LBD.

Table 4.1: Dominating cost (in flops) of each stage of the implicit LBDR restart

Stage of Method	Flops Required
k LBDR iterations	$2k^2(m+n) + 2\alpha mk$
Implicit restart (obtain $\hat{B}_{k-1}$)	$6k(m+n)$
a LBDR iteration (obtain $\hat{B}_k$)	$2k(m+n) + 2\alpha m$

We note that Demmel and Kahan [5] have shown that when $\mu = 0$ the work in one QR iteration (i.e., the work involved in forming $\hat{B}_k$ in (4.4)) can be reduced. Their zero shift QR algorithm uses no addition and subtractions, and has the property that it computes each entry of the transformed matrix to high *relative* accuracy.

We note that the LBDR factorization of dimension k with p restarts can be computed in several ways, since mathematically restarts and Lanczos steps commute. These options will also in practice yield equivalent results. Since the cost of a restart increases with k, it follows that it is advantageous to perform necessary restarts as early as possible.

In the following, we will be interested in interspersing restarts into LBDR. One could, for example, perform an implicit restart after every standard LBDR iteration. The amount of computations required in this case to obtain a decomposition of size k doubles that of LBDR alone (assuming $k \approx \alpha$). However, note that the work involved in LBDR grows quadratically in k and that properly applied restarts will typically reduce the number of iterations needed to obtain a satisfactory solution. We stress that even a modest reduction in k can more than compensate for the extra effort required by implicit restarts.

4.3 Comparing explicit and implicit restarts

The Lanczos algorithm, in contrast to the power method, typically converges to numerous singular values without breaking down after finding an invariant subspace of low dimension. However, restarting (multiplying b by shifted versions of AA^T) does introduce a power method aspect into the algorithm. The starting vector $\hat{u}_1 = \hat{\beta}_1^{-1}(AA^T)^p b$ will converge to the singular vector, $u_{\sigma 1}$, as p grows. But as a result (recall (2.1) and (2.2)),

$$\hat{\alpha}_1 = \sqrt{\hat{u}_1^T AA^T \hat{u}_1} \approx \sigma_1 \sqrt{\hat{u}_1^T u_{\sigma 1}} \approx \sigma_1,$$
$$\hat{\beta}_2 \hat{u}_2 = \hat{\alpha}_1^{-1} AA^T \hat{u}_1 - \hat{\alpha}_1 \hat{u}_1 \approx (\hat{\alpha}_1^{-1}\sigma_1^2 - \hat{\alpha}_1) u_{\sigma 1} \approx 0.$$

The fact that $\hat{u}_1 \approx u_{\sigma 1}$ yields a catastrophic cancellation in the calculation of $\hat{\beta}_2 \hat{u}_2$. This loss of precision will not force a complete breakdown of the Lanczos algorithm; but the results of explicitly restarted LBDR can deviate significantly from what is expected through theory.

Implicit restarts, on the other hand, do not exhibit such behavior. In an implicit restart $\hat{u}_2$ is not formed from $\hat{u}_1$. Rather, $\hat{u}_2 U_{k+1} \hat{Q}_l e_2$ is computed by an orthogonal rotation of the columns of U_{k+1}, which explains why implicit restarts do not entail the loss of precision encountered by explicit restarts.

As an example in the context of ill-posed problems, we again consider the Fredholm integral equation of (3.10), discretized by collocation with piecewise linear splines and nodes $s_j = j/n, n = 50$. Following Hanke [11], the data for this example is taken as

$$g(s) = (7s^6 - 18s^5 + 15s^4 - 5s^3 + s)/3 + \epsilon, \tag{4.12}$$

where ϵ is normally distributed $\mathcal{N}(0, 10^{-3})$.

Given the SVD of the full matrix $A \in R^{50 \times 50}$, the GCV function takes on a minimum at $\nu = 7$. To obtain an solution based on approximations to the 7 largest singular values and vectors of A we will examine two modified LBDR approaches. In the first approach, 7 LBDR steps are explicitly performed with the starting vector $(AA^T)^4 b$. The resulting bidiagonal matrix is then used (i.e., see (2.8)) to compute an approximate solution, $\hat{x}_{7_{expl}}$. In the second approach, 11 LBDR iterations are performed with the starting vector b. Four implicit restarts with zero shifts are then performed to yield a bidiagonal matrix and a corresponding approximate solution $\hat{x}_{7_{impl}}$. The explicit and implicit approaches should produce identical results in exact arithmetic. However, although all computations were performed in double precision, Table 4.2 indicates that the explicitly computed solution is clearly inferior to the implicit one, $\hat{x}_{7_{impl}}$. Note that in Table 4.2, $\bar{x}$ is the exact solution to the noiseless problem with $\epsilon = 0$ in (4.12). The computed solution, $x_7 = \sum_{i=1}^{7}(u_{\sigma i}^T b v_{\sigma i})/\sigma_i$, is in a sense the best obtainable solution as it is based on exact singular values and vectors of A.

Table 4.2: Explicit vs. Implicit Restart Methods for the Ill-posed Problem

Approach	Error Expression	Error
Truncating exact SVD data	$\frac{\|\bar{x}-x_7\|}{\|\bar{x}\|}$	0.0247
Explicitly restarted LBDR	$\frac{\|\bar{x}-\hat{x}_{7_{expl}}\|}{\|\bar{x}\|}$	0.0529
Implicitly restarted LBDR	$\frac{\|\bar{x}-\hat{x}_{7_{impl}}\|}{\|\bar{x}\|}$	0.0250

As one might expect, the implicitly restarted solution, $\hat{x}_{7_{impl}}$ agrees well with the "best" obtainable solution, x_7. The error corresponding to $\hat{x}_{7_{expl}}$, on the other hand, is twice as high as the error for the other two methods. In the explicit restart, the quantity $\hat{\beta}_2\hat{u}_2$ is small enough ($\|\hat{\beta}_2\hat{u}_2\| = 2.721 \cdot 10^{-6}$) to introduce a loss of precision into LBDR and thus also into $\hat{x}_{7_{expl}}$.

5 Regularization with implicit restarts

It was demonstrated in §3.3 that the combination of non-restarted LBDR and GCV could not be treated as a black box method for solving (1.1). Unless an acceptable heuristic weighting for the GCV function can somehow be determined, the number of iterations, k, may grow large before a reliable estimation of the optimal regularization parameter, ν_*, is achieved.

Implicit restarts are incorporated into LBDR in this section in order to improve the determination of the optimal regularization parameter through better approximations of the SVD of A. As a result, examples demonstrate that the value of calculated for ν_* via the restarted LBDR GCV curve is a reliable substitute for the parameter derived from the full-information GCV curve computed from the SVD of A. Moreover, restarted LBDR finds the minimizing value of the full-information GCV curve with k only slightly larger than optimal and without extra weightings on the LBDR GCV curve.

5.1 Implementation

In implementing implicit restarts into regularized LBDR, our task is to acquire from $G(k, \nu)$ an estimate for the optimal regularization parameter, ν_*, which corresponds to the parameter computed from the full-information GCV curve. In particular, false minimums on the LBDR GCV curve (recall the example in §3.3) are to be eliminated.

To achieve these goals, two issues must first be addressed: (1) when should zero-shift implicit restarts be applied and (2) how large should the value of k be. It should be stressed that no definite solutions exist for these questions. Rather, as in all existing hybrid iterative methods, we use well justified heuristics.

Our approach to issue one concentrates on the minimums of the the LBDR GCV curves. Specifically, restarts are employed to insure that the global minimum of $G(k,\nu)$ at $\nu = \nu_g$ corresponds to a minimum of $G(\nu)$ at $\nu = \nu_g$. Implicit restarts with zero shifts force the convergence of $\hat{B}_k$'s largest singular values which in turn drive the $G(k,\nu)$ curve to the $G(\nu)$ curve for values of ν less than k. Restarts insure that the global minimum of $G(k,\nu)$ is legitimate. The convergence of the LBDR GCV curve at ν_g to the full-information curve is determined by examining the effects of restarts on the values $G(k,\nu)|_{\nu=\nu_g}$ and $G(k,\nu)|_{\nu=\nu_\ell}$ where ν_ℓ is the location of the smallest localized minimum on the $G(k,\nu)$ curve. Convergence is assumed when the changes in $G(k,\nu)$ at ν_g and ν_ℓ become progressively smaller than the difference between the two values, $G(k,\nu)|_{\nu=\nu_\ell} - G(k,\nu)|_{\nu=\nu_g}$.

Restarts are utilized to drive the global minimum of $G(k,\nu)$ towards the minimum of $G(\nu)$ over $1 \leq \nu < k$. Thus one expects to see the restarted LBDR/GCV estimate of ν_* converge to that of the full-information GCV case when k is only slightly larger than the value of ν which minimizes $G(\nu)$. The convergence of $G(k,\nu)$'s estimate for ν_* (i.e., the appearance of the same value of ν_{global} for several consecutive values of k) provides an answer to the second question: how large should k become. Note that this stopping criterion, the convergence of v_g as k grows, is in practice equivalent to the convergence of x_{k,ν_g} since the larger singular values and vectors of A are obtained quickly with restarted LBDR. The use of the convergence of $x_{k,\nu}$ as a stopping criterion is discussed in [12].

Given the above discussion, a restarted, regularized LBDR routine can now be implemented. This implementation consists of two *while* loops. The inner loop locates a legitimate minimum of $G(k,\nu)$ for a fixed value of k. The outer loop compares the values of ν_g as k grows in order to determine when convergence occurs. The converged value of ν_g serves as the estimate for ν_*.

ALGORITHM 5.1. *Implicitly restarted LBDR ill-posed system solver*

```
1. Initialize $u_1 = b/\beta_1$ and $k = 1$.
2. While the estimates for the regularization parameter, $\nu_*$, have not converged,
    (a) perform one LBDR iteration: $[U_{k+1}, V_k, B_k] \leftarrow LBDR(U_k, V_{k-1}, B_{k-1})$.
    (b) compute the SVD of $B_k$ and the GCV function, $G(k,\nu)$.
    (c) determine the values, $\nu$, which are the global and possibly local minimizers
        of $G(k,\nu)$, i.e., $\nu_g$ and $\nu_\ell$.
    (d) while $\nu_\ell$ exists and while $G(\nu,k)|_{\nu=\nu_g}$ has not converged,
        i. perform an implicit restart with $\mu = 0$:
           $[\hat{U}_k, \hat{V}_{k-1}, \hat{B}_{k-1}] \leftarrow restart(U_{k+1}, V_k, B_k, 0)$.
```

```
            ii. recover the last LBDR iteration:
                $[U_{k+1}, V_k, B_k] \leftarrow LBDR(\hat{U}_k, \hat{V}_{k-1}, \hat{B}_{k-1})$.
            iii. compute a new GCV curve
            iv. determine new values for $\nu_g$ and possibly $\nu_\ell$.
        (e) set the $k^{th}$ estimate for $\nu_*$ equal to $\nu_g$.
        (f) $k \leftarrow k+1$.
    3. Compute the regularized solution, $x_{k,\nu_g}$, via (2.15).
```

There are a few finer details in the implementation which still require additional comments. First, note that implicit restarts are only applied when the curve $G(k,\nu)$ possesses multiple minimums. This rule is a safe yet efficient heuristic which keeps unnecessary implicit restarts from being performed too early in the algorithm. It is safe because a minimum corresponding to the desired regularization parameter must always eventually appear (recall Figure 3.1 where $k = 20$, $v_{local} = 9$ and $v_{global} = 20$). When multiple minimums exist, restarts are used to locate the legitimate one.

Using the terminology of §3.3, it should also be noted that the GCV curves of steps (2.c) and (2.e.iii) are weighted with δ fixed at one. To avoid estimating the variance of the noise on b using only one piece of data, the value $G(k,\nu)|_{\nu=k}$ is not taken into consideration. We stress that this weighting is independent of the problem. Furthermore, this weighting is not required; however it does make a significant impact in practice.

5.2 Numerical examples

To investigate the performance of implicitly restarted LBDR, we return to the problem deriv2 which was introduced in §3.3. So that comparisons can made, the performance of non-restarted LBDR will also be studied. For the non-restarted LBDR, the weighting $\delta = 3$ is applied to the GCV function.

In order to be as unbiased as possible, the experiment will determine the number of steps, k, each approach takes before their respective estimates for ν_* converge to the value of ν which minimizes the full-information GCV curve. The assumption that the minimum of $G(v)$ is known a priori cannot be met in practice. But for the purpose of this example, this assumption allows us to avoid the heuristics involved in determining when an estimate for ν_* has converged (e.g., for how many consecutive values of k must v_g remain constant in step 1 of Algorithm 5.1 before convergence is assumed). We will, however, briefly return to this practical problem in the concluding remarks of §6.

Figure 5.1 plots the number of iterations the restarted and non-restarted LBDR approaches take to converge to the full-information estimate for the optimal regularization parameter. Experiments were performed with fifty different right hand sides. The noise on each b is normally distributed $\mathcal{N}(0, 10^{-4})$, and corresponds to consecutive random generator seeds starting from 0. On average, the restarted LBDR required 2.16 less steps than non-restarted LBDR to compute the full-information estimate for ν_*. However, the restarted case required an average of 3.30 implicit restarts. Note that the restarted LBDR consistently converges with k only slightly larger than the full-information estimate for ν_*. However, the restarted LBDR is slightly more expensive in this case.

Figure 5.2 plots the number of iterations needed when the noise on the b is normally distributed $\mathcal{N}(0, 10^{-5})$. With this level of noise, restarted LBDR converges with an average of 8.02 less steps but with an average cost of 7.28 implicit restarts. Recall that each implicit restart is two to three times as expensive as a simple LBDR iteration for a fixed k. However, the amount of work in non-restarted LBDR grows quadratically in k. Thus the work corresponding to the extra iterations required by non-restarted LBDR approximately cancels out the cost of the restarts. From a storage standpoint, restarted LBDR is clearly preferred.

As the noise level drops in this problem, restarted LBDR will overtake its non-restarted counterpart. Increasing the weighting, δ, in non-restarted LBDR will help remove the "spikes" in Figure 5.2; but not without a price. Underestimating the weighting (e.g., $\delta = 1$), on the other hand, will cause non-restarted LBDR to fail for any level of the noise on b. Regardless of the noise level, restarted LBDR consistently converges with k only slightly larger than the full-information estimate for the optimal regularization parameter.

For a second example, we return to the problem of Hanke [11] which was defined in §4.3. Throughout this problem, the noise level will be normally distributed $\mathcal{N}(0, 10^{-4})$. The number of nodes, s_j, used to discretize the problem will be varied though. Figure 5.3 displays the number of restarted and non-restarted LBDR iterations required to converge to the full-information estimate for ν_* when 50 nodes are used. Restarted LBDR converges on average with 4.54 fewer iterations than non-restarted LBDR but with an average cost of 3.66 implicit restarts. Figure 5.4 displays the number of restarted and non-restarted LBDR iterations needed for convergence when 100 nodes are used. In this case, restarted LBDR converges on average with 7.46 fewer iterations but with an average cost of 3.70 implicit restarts.

For the fifty node case, the restarted and non-restarted methods finish about the same in terms of total computations. But as the number of nodes increases, restarted LBDR begins to pull ahead. We stress that restarted LBDR performs robustly as the noise and/or the problem is varied.

We remark on the fact that no error comparisons were made between the approximate solutions, $x_{k,\nu}$, and the actual solution, $\bar{x}$, in the examples above. The reason for leaving out this data is that the approximate solutions depend primarily on the different methods' (TSVD's, non-restarted LBDR's, and restarted LBDR's) estimates for ν_*. Here these estimates were fixed as the minimum of $G(\nu)$ so that a fair comparison of restarted and non-restarted LBDR could be achieved. Providing plots of the approximate solutions versus $\bar{x}$ would comment more on the effectiveness of the GCV curve than on the value of restarts. In practice, the full information GCV's estimate for ν_* cannot be known a priori.

Even if the issue of weighting the $G(k, \nu)$ curve can be avoided by restarts, one must be concerned with determining when the LBDR/GCV estimate for ν_* has converged. Restarted LBDR will often temporarily converge to the first significant local minimum

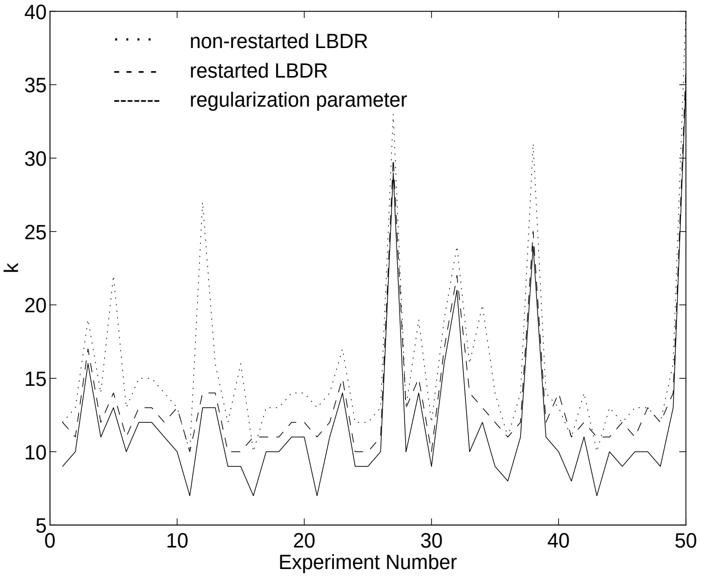

Figure 5.1: Number of iterations needed for restarted (dashed line) and non-restarted (dotted line) LBDR to converge to the full-information estimate of ν_* (solid line) for the problem deriv2 when the noise is normally distributed $\mathcal{N}(0, 10^{-4})$.

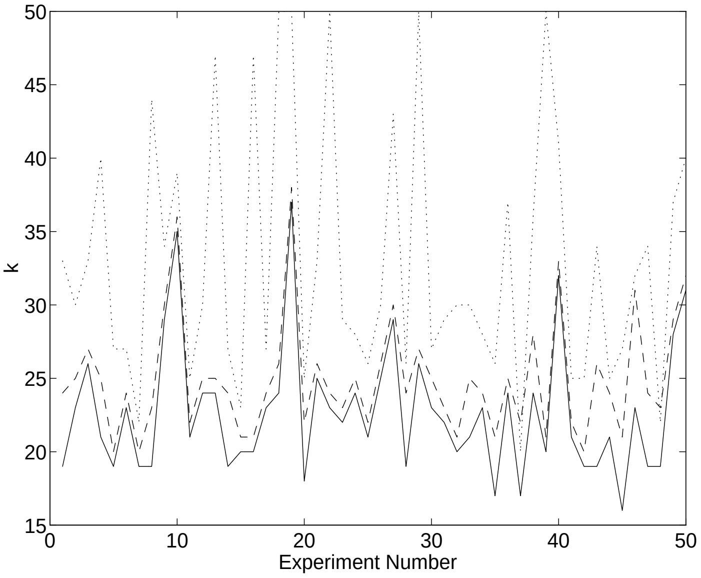

Figure 5.2: Number of iterations needed for restarted (dashed line) and non-restarted (dotted line) LBDR to converge to the full-information estimate of ν_* (solid line) for the problem deriv2 when the noise is normally distributed $\mathcal{N}(0, 10^{-5})$.

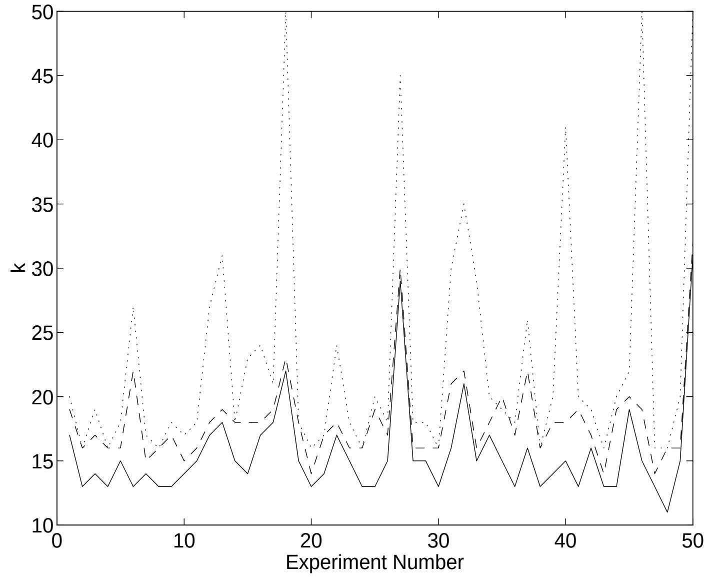

Figure 5.3: Number of iterations needed for restarted (dashed line) and non-restarted (dotted line) LBDR to converge to the full-information estimate of ν_* (solid line) for the problem of [hank:91] discretized with 50 nodes.

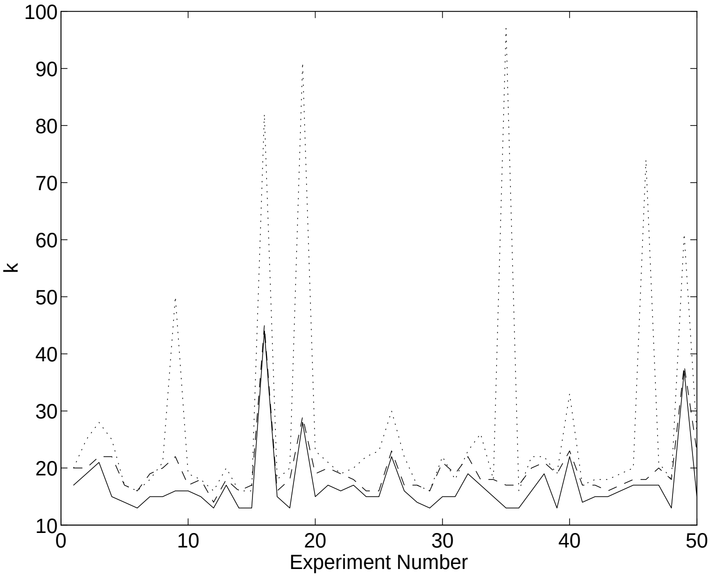

Figure 5.4: Number of iterations needed for restarted (dashed line) and non-restarted (dotted line) LBDR to converge to the full-information estimate of ν_* (solid line) for the problem of [hank:91] discretized with 100 nodes.

of the full-information GCV curve before eventually converging to the global minimum (note that the GCV curve usually has many minimums which may only slightly differ from each other). Practical experience suggests that it is better to under-estimate rather than over-estimate the regularization parameter. As a result, it is typically better to choose the regularization parameter to correspond to the first significant local minimum on the GCV curve instead of the global minimum.

6 Concluding Remarks

For many large discrete ill-posed least squares problems, the conjugate gradient method applied to the normal equations (CGLS) often converges rapidly to a nearly optimal solution. A serious drawback of this method is that no reliable stopping criterion for CGLS is known. This is true in particular when a priori information about the noise level is lacking.

In this paper we have considered methods based on Lanczos bidiagonalization with full reorthogonalization (LBDR). Although these are more costly than CGLS in terms of operations and storage, they allow the use of more reliable stopping criteria, e.g., generalized cross validation (GCV). In particular we have developed and tested an LBDR/GCV algorithm which uses implicit restarts with zero shifts, corresponding to the use of starting vectors of the form $(AA^T)^p b$. Our limited numerical experience with this algorithm indicates that it is a numerically stable and robust iterative method for solving large discrete ill-posed systems. A major advantage is that the regularization parameter computed by implicitly restarted LBDR/GCV will converge to the parameter of the full-information, TSVD/GCV case with the dimension k of the Krylov space being only slightly larger than TSVD/GCV's estimate for ν_*.

As with all regularization methods, the success of implicitly restarted LBDR/GCV depends to a large extent upon the type of problem being solved. Our work with restarted LBDR/GCV suggests that is better suited problems in which the solution coefficients fall gradually to zero. For other classes of problems, the use of the L-curve criterion [13] instead of GCV may be preferred. The merits of an implicitly restarted LBDR/L-curve method remain an area for further research.

REFERENCES

1. Å. Björck *A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations.* BIT, 28 (1988), pp. 659–670.

2. Å. Björck *Numerical Methods for Least Squares Methods.* Frontier in Applied Mathematics series. SIAM, Philadelphia, to appear.

3. H. Brakhage. *On ill-posed problems and the method of conjugate gradients.* In Inverse and Ill-Posed Problems, H. W. Engl, C. W. Groetsch. eds, Academic Press, Boston, New York, London, pp. 165–175, 1987.

4. D. Calvetti, L. Reichel, and D. C. Sorensen. *An implicitly restarted Lanczos method for large symmetric eigenvalue problems.* Elect. Trans. Numer. Anal., 2 (1994), pp. 1—21.

5. J. Demmel and W. Kahan. *Accurate singular values of bidiagonal matrices.* SIAM J. Sci. Statist. Comput., 11 (1990), pp. 873–912.

6. D. A. Girard. *A fast 'Monte-Carlo Cross-Validation' procedure for large least squares problems with noisy data.* Numer. Math. 56 (1989), pp. 1–23.

7. G. H. Golub, M. T. Heath, and G. Wahba. *Generalized cross-validation as a method for choosing a good ridge parameter.* Technometrics, 21 (1979), pp. 215–223.

8. G. H. Golub and W. Kahan. *Calculating the singular values and pseudo-inverse of a matrix.* SIAM J. Numer. Anal. Ser. B, 2 (1965.), pp. 205–224.

9. G. H. Golub, F. T. Luk, and M. L. Overton. *A block Lanczos method for computing the singular values and singular vectors of a matrix.* ACM Trans. on Math. Soft., 7 (1981), pp. 149-169.

10. G. H. Golub and C. F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, second edition, 1989.

11. M. Hanke. *Accelerated Landweber iterations for the solution of ill-posed equations.* Numer. Math. 60 (1991), pp. 341–373.

12. M. Hanke and P. C. Hansen. *Regularization methods for large-scale problems.* Surveys on Mathematics for Industry, 3 (1994), pp. 253–315.

13. P. C. Hansen. *Analysis of discrete ill-posed problems by means of the L-curve.* SIAM Review, 34 (1992), pp. 561–580.

14. P. C. Hansen. *Regularization tools: a MATLAB package for analysis and solution of discrete ill-posed problems.* Numerical Algorithms, 6 (1994), pp. 1–36.

15. A. K. Louis *Convergence of the conjugate gradient method for compact operators.* In Inverse and Ill-Posed Problems, H. W. Engl, C. W. Groetsch. eds, Academic Press, Boston, New York, London, pp. 177–183, 1987.

16. D. P. O'Leary and J. A. Simmons. *A bidiagonalization-regularization procedure for large scale discretizations of ill-posed problems.* SIAM J. Sci. Statist. Comput., 2 (1981), pp. 474–489.

17. C. C. Paige. *Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix.* J. Inst. Math. Applics., 18 (1976), pp. 341–349.

18. C. C. Paige and M. A. Saunders. *LSQR an algorithm for sparse linear equations and sparse least squares.* ACM Trans. Math. Software, 8 (1982), pp. 43–71.

19. D. Sorensen. *Implicit application of polynomial filters in a k-step Arnoldi method.* SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.

20. C. R. Vogel and J. G. Wade *Iterative SVD-based methods for ill-posed problems.* SIAM J. Sci. Statist. Comput., 15 (1994), pp. 736–754.

21. G. Wahba. *Spline Models for Observational Data.* SIAM, CBMS 59, 1990.

22. J. M. Varah. *A practical examination of some numerical methods for linear discrete ill-posed problems.* SIAM Review, 21 (1979), pp. 100–111.