Abstract: Consider the solution of the linear system $Ax = b$, where A is nearly singular. The solution x can be uniquely decomposed into two parts: a generally large component in the direction of the approximate null space of A, and a part that is orthogonal to it. In many applications, it is desirable to compute this *deflated decomposition* in a stable and efficient manner. In this paper, we propose an iterative algorithm based on the Lanczos process, for the case where A is symmetric. The method requires access to A only in the form of a matrix-vector product Av and is efficient for large problems. No a priori knowledge of the approximate null space is needed.

Deflated Lanczos Procedures for Solving Nearly Singular Systems

Tony F. Chan and Youcef Saad

Research Report YALEU/DCS/RR-403
October 1985

The authors were supported in part by the Department of Energy under contract DE-AC02-81ER10996 and by the Army Research Office under contract DAAG-83-0177.

Keywords: *Deflated decomposition, nearly singular systems, the Lanczos method, conjugate gradient, Krylov subspace, QL method.*

Abstract: Consider the solution of the linear system $Ax = b$, where A is nearly singular. The solution x can be uniquely decomposed into two parts: a generally large component in the direction of the approximate null space of A, and a part that is orthogonal to it. In many applications, it is desirable to compute this *deflated decomposition* in a stable and efficient manner. In this paper, we propose an iterative algorithm based on the Lanczos process, for the case where A is symmetric. The method requires access to A only in the form of a matrix-vector product Av and is efficient for large problems. No a priori knowledge of the approximate null space is needed.

Deflated Lanczos Procedures for Solving Nearly Singular Systems

Tony F. Chan and Youcef Saad

Research Report YALEU/DCS/RR-403
October 1985

The authors were supported in part by the Department of Energy under contract DE-AC02-81ER10996 and by the Army Research Office under contract DAAG-83-0177.

Keywords: *Deflated decomposition, nearly singular systems, the Lanczos method, conjugate gradient, Krylov subspace, QL method.*

1. Introduction

The problem we consider is the solution of the linear system

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

where $A \in R^{N \times N}$ is symmetric positive definite but possibly nearly singular. We will denote the eigenvalues of A by $\lambda_1 \leq \lambda_2 \leq \ldots, \leq \lambda_N$, and the corresponding orthonormal eigenvectors by $\{w_1, w_2 \ldots, w_N\}$. For simplicity, we will assume that the near-nullity of A is at most one, i.e., that no eigenvalue other than λ_1 could be close to zero. The positive definiteness assumption of A can be slightly relaxed, by replacing it with the less restrictive assumption that a few of the first eigenvalues of A are negative.

The solution to (1.1) can be expressed in the form:

$$x = x_d + \frac{w_1^T b}{\lambda_1} w_1, \tag{1.2}$$

where

$$x_d \equiv \sum_{i=2}^{N} \frac{w_i^T b}{\lambda_i} w_i.$$

In the above expression, we have separated the part of x in the direction of w_1 from the part x_d that is orthogonal to it. The vector x_d is called the *deflated solution* of (1.1) and (1.2) is called the *deflated decomposition* of x. When λ_1 is small but $w_1^T b$ is not small, then the last term in (1.2) will dominate x_d, and in finite precision arithmetic, it would be difficult to recover x_d from x with close to full machine precision. Therefore, when A is nearly singular, it is often more appropriate to compute the deflated decomposition of x rather than to compute x directly. This requires computing x_d, λ_1 and w_1.

The deflated decomposition is useful in many applications, such as when solving bordered singular systems [5, 4, 1] that arise in continuation methods for solving nonlinear systems [3, 9] and bifurcation computations [8, 15] and in constrained optimization problems[7].

Of course, one could compute the deflated decomposition by first computing the eigenvalue decomposition of A, but this is often too expensive for large problems. In earlier work by Stewart [20] and Chan [2], the deflated decomposition is computed by *implicit* algorithms in which only a direct solver for A (such as an LU factorization method) is needed. In this paper, we look at how iterative methods based on the Lanczos process can be used to compute the deflated decomposition in a numerically stable manner. The objective is to be able to compute the deflated decomposition by only accessing A in the form of a matrix-vector product Av. Such a method has obvious advantages for large problems. Basically, the Lanczos method produces a small tridiagonal matrix T approximately similar to A and the deflation techniques in [2, 20] can then be applied to T. In addition, we propose a new method based on the QL iteration for deflating the tridiagonal matrice T.

Note that, as explained above, the naive method of applying a Lanczos type procedure, such as the symmetric conjugate gradient method, to (1.1) directly will fail to compute x_d accurately when A is nearly singular. If w_1 is known *a priori*, then a modified conjugate gradient algorithm, in which the iterate at each iteration is orthogonalized with respect to w_1 [10], might be expected to be effective. The algorithms to be presented in this paper do not require *a priori* knowledge of w_1 or λ_1.

In Section 2, we review the deflation techniques of [2, 20] and present the new QL algorithm for tridiagonal matrices. In Section 3, we review the Lanczos process and then in Section 4, we explain how to combine the techniques of Sections 2 and 3 to compute the deflated decomposition of (1.1). In Section 5, we discuss some implementation issues concerning the loss of orthogonality of the Lanczos vectors and stopping criteria. In Section 6, we present a convergence analysis that essentially shows that the deflated solution converges at the rate that would have been achieved if we removed the eigenvalue λ_1 from the spectrum of the original matrix A. Finally, we present some numerical results in Section 7 and end in Section 6 with some concluding remarks. Throughout this paper, upper case Latin letters denote matrices, lower case Latin letters denote vectors and lower case Greek letters denote scalars. We will use the notation $\|\cdot\|$ to denote the 2–norm.

2. Deflated Decomposition for Tridiagonal Systems

In this section, we discuss techniques for computing the deflated decomposition of solutions to the linear system

$$Tz = f, \tag{2.1}$$

where T is tridiagonal. First, we review the deflation techniques of [2, 20], which are based on orthogonal projections. These techniques are designed for general linear systems but can be applied to tridiagonal systems to produce an efficient deflation algorithm. Next we will present a new deflation algorithm based on the QL iteration specifically designed for tridiagonal matrices.

2.1. Deflation by Orthogonal Projection

Let $P \equiv I - u_1 u_1^T$ denote the orthogonal projector with respect to the eigenvector u_1 corresponding to the smallest eigenvalue λ_1 of T. Then the deflated solution z_d of $Tz = f$ can be characterized as the *unique* solution to the following singular but consistent system with a constraint:

$$PTz_d = Pf$$
$$Pz_d = z_d.$$

Based on this characterization, Stewart [20] and Chan [2] propose the following implicit algorithm for computing z_d.

Algorithm Deflate:

1. Compute λ_1 and u_1 of T by a few steps of inverse iteration.
2. Solve the system $T\hat{z} = Pf$ for $\hat{z}$.
3. Compute $z_d = P\hat{z}$.

It is shown in [2] that Algorithm Deflate computes z_d in a stable manner. For tridiagonal matrices, the above algorithm can take full advantage of efficient tridiagonal solvers.

2.2. Deflation by the QL Method

Assume that we have computed the eigenpair (λ_1, u_1) of the tridiagonal matrix T by inverse iteration. The main idea behind the QL-deflation method is that if we apply one step of the QL iteration [14] for computing the eigenvalues of T with the shift λ_1, then the resulting transformed tridiagonal matrix decouples in a way which allows the deflated decomposition to be computed easily and in a stable way.

Specifically, the QL transformation amounts to first computing the LQ-factorization of $T - \lambda_1 I$,

$$T - \lambda_1 I = LQ, \tag{2.2}$$

where L is lower triangular and Q is orthogonal, and performing the product in reverse order, adding back the shift:

$$T^{(1)} = QL + \lambda_1 I. \tag{2.3}$$

The transformation from T to $T^{(1)}$ is the well known QL-transformation with shift λ_1 [14]. In practical implementations the two operations (2.2) and (2.3) are performed in one single pass. It can be shown that $T^{(1)}$ has the form shown below

$$T^{(1)} = \left[\begin{array}{c|cccc} \lambda_1 & 0 & 0 & \cdots & 0 \\ \hline 0 & & & & \\ 0 & & & & \\ & & & \widehat{T} & \\ \vdots & & & & \\ 0 & & & & \end{array}\right]$$

where $\widehat{T}$ is tridiagonal. It is well known that $T^{(1)}$ is unitarily similar to T since

$$T^{(1)} = QL + \lambda_1 I = Q(T - \lambda_1 I)Q^T + \lambda_1 I = QTQ^T.$$

The system $Tz = f$ can now be transformed into one for $T^{(1)}$, namely,

$$T^{(1)}y = Qf \equiv \widehat{f}, \tag{2.4}$$

where $y = Qz$. If we partition y and $\widehat{f}$ into $(y_1, y_2)^T$ and $(f_1, f_2)^T$ to conform with the partition of $T^{(1)}$, then the deflated solution y_d of (2.4) is easily seen to be:

$$y_d = (0, \widehat{T}^{-1} f_2)^T.$$

The deflated solution z_d of (2.1) can then be obtained by

$$z_d = Q^T y_d.$$

Since $\widehat{T}$ is nonsingular by the assumption that the nullity of A is at most one, this approach requires only solutions of nonsingular tridiagonal systems.

3. The Lanczos Algorithm for Solving Linear Systems.

In this section we put aside the issue of near singularity temporarily and describe the Lanczos method for solving linear systems [13]. Consider the linear system:

$$Ax = b \tag{3.1}$$

Suppose that a guess $x^{(0)}$ to the solution is available and let r_0 be its residual vector: $r_0 = b - Ax^{(0)}$. Then the Lanczos algorithm for solving (3.1) can be described as follows:

Algorithm: The Lanczos algorithm for solving linear systems

```
(1) Stage 1 : Generate the Lanczos Vectors
    • Start: $v_1 = r_0/||r_0||$
    • For $j = 1, 2 \ldots m$ compute
        $\alpha_j := (Av_j, v_j)$    (3.2)
        $\hat{v}_{j+1} := Av_j - \alpha_j v_j - \beta_j v_{j-1}, \quad (\beta_1 v_0 \equiv 0)$    (3.3)
        $\beta_{j+1} := ||\hat{v}_{j+1}||$    (3.4)
        $v_{j+1} := \hat{v}_{j+1}/\beta_{j+1}$    (3.5)
(2) Stage 2 : Form the approximate solution
        $x^{(m)} := x^{(0)} + V_m T_m^{-1}(||r_0||e_1)$    (3.6)
    where $V_m = [v_1, v_2, ..v_m]$ and $T_m$ is the tridiagonal matrix $Tridiag[\beta_j, \alpha_j, \beta_{j+1}]$.
```

In theory, the vectors v_i computed from stage 1 of this process form an *orthonormal basis* of the Krylov subspace $K_m = span\{r_0, Ar_0, ..A^{m-1}r_0\}$. It can be easily verified that

$$AV_m = V_m T_m + \beta_{m+1} v_{m+1} e_m^T \tag{3.7}$$

and therefore that $V_m^T A V_m = T_m$ which means that T_m is nothing but the matrix representation of the section of A in the Krylov subpace K_m with respect to the basis V_m. Furthermore, it is easily seen that the Lanczos algorithm realizes a projection process, i.e., a Galerkin process, onto the Krylov subspace K_m [13, 16]. In other words the approximate solution $x^{(m)}$ can be found by expressing that it belongs to the affine subspace $x^{(0)} + K_m$ and that its residual vector $b - Ax^{(m)}$ is orthogonal to K_m. Denoting by Π_m the orthogonal projector onto K_m, this means that the Lanczos method solves the approximate problem:

$$Find \;\; x^{(m)} \in x^{(0)} + K_m \;\; such \;\; that :$$

$$\Pi_m(b - Ax^{(m)}) = 0 \tag{3.8}$$

The approximation thus computed is identical with that provided by m steps of the conjugate gradient (CG) method when A is positive definite [13]. When A is not positive definite this relation between the Lanczos algorithm and the CG method can be exploited to derive stable generalizations of the CG algorithm to symmetric indefinite systems [12, 13, 6, 17]. In this paper we will show another way of exploiting this relation to provide a method for treating nearly singular systems, which can be viewed as a variation of the conjugate gradient method.

4. The Lanczos Deflation Algorithm

According to the previous section, when A is not nearly singular the approximate solution to (1.1) can be obtained by a Lanczos - conjugate gradient method with the solution given by:

$$x^{(k)} = x^{(0)} + V_k z^{(k)}$$

where $z^{(k)}$ is a k-dimensional vector which satisfies the equation

$$T_k z^{(k)} = ||r_0|| e_1. \tag{4.1}$$

When A is nearly singular, a difficulty arises in the solution of the above tridiagonal system. Indeed for large k it is well-known that the extreme eigenvalues of T_k will converge to the corresponding extreme eigenvalues of A, see e.g. [14], and as a result the eigenvalues closest to zero of the matrix T_k for large enough k will be close to the eigenvalues closest to zero of A, i.e., it will be just as nearly singular.

The obvious remedy is to solve (4.1) with either of the two deflation procedures of Section 2. Let us assume that we compute the smallest eigenvalue $\lambda_1^{(k)}$ and the associated eigenvector $u_1^{(k)}$ of the tridiagonal matrix T_k by inverse iteration. If we apply the deflation techniques of Section 2 to the tridiagonal system (4.1), we obtain the decomposition

$$z^{(k)} = z_d^{(k)} + \frac{\gamma^{(k)}}{\lambda_1^{(k)}} u_1^{(k)} \tag{4.2}$$

where $\lambda_1^{(k)}$ is the smallest eigenvalue of T_k, $u_1^{(k)}$ is the eigenvector associated with $\lambda_1^{(k)}$, $z_d^{(k)}$ is the deflated solution of (4.1) and

$$\gamma^{(k)} = ||r_0|| e_1^T u_1^{(k)}.$$

Multiplying both members of the above equation on the left by the matrix V_k and adding $x^{(0)}$ we obtain

$$x^{(k)} = x^{(0)} + V_k z^{(k)} = x^{(0)} + V_k z_d^{(k)} + \frac{\gamma^{(k)}}{\lambda_1^{(k)}} V_k u_1^{(k)}. \tag{4.3}$$

Observe that in the above equation the vector $V_k u_1^{(k)}$ is nothing but the Ritz vector associated with the eigenvalue of A closest to zero, i.e., it is the approximate eigenvector of A computed by the Lanczos process from the Krylov subspace K_k. We will denote it by $w_1^{(k)}$. Hence we obtain an approximate deflated solution in the form

$$x^{(k)} = x_d^{(k)} + \frac{\gamma^{(k)}}{\lambda_1^{(k)}} w_1^{(k)}, \tag{4.4}$$

where

$$x_d^{(k)} = x^{(0)} + V_k z_d^{(k)}. \tag{4.5}$$

We point out the important fact that (4.4) is the orthogonal decomposition of the vector $x^{(k)} - x^{(0)}$ in the direction $w_1^{(k)}$ in the subspace K_k. This is true because, as may be readily shown, $V_k z_d^{(k)}$ is orthogonal to the Ritz vector $w_1^{(k)}$.

Note that we need to store the vectors v_i as they are generated and retrieve them once when forming the approximation $x_d^{(k)}$.

5. Practicalities

5.1. Loss of Orthogonality of the Lanczos Vectors

A troublesome behavior of the Lanczos algorithm is the loss of orthogonality of the vectors $v_i, i = 1, \dots m$. Fortunately, this does not prevent the method from converging but often results in a slow-down. Parlett, Scott and Simon [13, 18, 19] have proposed several different practical reorthogonalization techniques. Loss of orthogonality is a phenomenon that cannot be avoided without some form of reorthogonalization but can be delayed by replacing the trivial implementation (3.2) – (3.5) by the following one [14, 11, 18]

$$\begin{aligned} q &:= Av_j - \beta_j v_{j-1} \\ \alpha_j &:= (q, v_j) \\ q &:= q - \alpha_j v_j. \end{aligned}$$

Then $\hat{v}_{j+1} \equiv q$ and (3.4) and (3.5) deliver the next vector v_{j+1}. This simply corresponds to a modified Gram-Schimdt step, instead of the regular Gram-Schmidt method of (3.2) - (3.4). Additional reorthogonalization can be added to further post-pone loss of orthogonality at little extra cost, by appending the following reorthogonalization steps to the above.

$$\begin{aligned} \delta &:= (q, v_{j-1}) \\ q &:= q - \delta v_{j-1} \\ \delta &:= (q, v_j) \\ \alpha_j &:= \alpha_j + \delta \\ q &:= q - \delta v_j \end{aligned}$$

Again define $\hat{v}_{j+1} := q$ and apply (3.4) and (3.5) to get v_{j+1}.

So far we have not discussed the possible negative effects of the loss of orthogonality in the Lanczos process mentioned earlier. An important factor of this phenomenon is the way in which it appears. Basically, loss of orthogonality is a signal that one or a few approximate eigenvalues of the matrix A have reappeared after they have already converged. As a result one can expect that as soon as the eigenvalue $\lambda_1^{(k)}$ has converged to λ_1 then a second copy of this eigenvalue will appear in the following steps of Lanczos. The presence of this extra eigenvalue can be disastrous if we solve the tridiagonal system without some extra precaution, because we are now facing a tridiagonal system with multiple sigularity.

As will be seen in Section 6 the first eigenvector will likely converge at the same time as the deflated solution converges, so this phenomenon will seldom hamper the progress of our procedure. However, as is discussed in section 5.2 a reasonable computational code should foresee hard cases that might occur such as when the smallest eigenvalue of A is so well separated from the others that convergence of the eigenvalue is very fast. There are two possible remedies to the problem of loss of orthogonality. The first one is to simply deflate more carefully in the tridiagonal system solution, i.e., to deflate by as many eigenvectors as there are small eigenvalues.

The second solution, which is well knowm in the context of the eigenvalue problem [14, 18] is to perform a selective reorthogonalization (SO) in the Lanczos process. Here, in contrast with full reorthogonalization, one only reorthogonalizes the current Lanczos vectors against the eigenvectors that have converged. Moreover, a clever implementation allows one to orthogonalize only when necessary. The idea of selective reorthogonalization is intuitively simple: since it is known that loss of orthogonality appears mainly in the direction of the converged Ritz vectors, then a remedy is to remove the corresponding components from the Lanczos vectors as soon as these start to reappear. The important point is that there are ways of determining when these components are likely to come back, for more details see [14, 19].

The above discussion suggests in fact that we may only have to reorthogonalize v_i against the converged Ritz vector that corresponds to the eigenvalue closest to zero.

5.2. Stopping Criteria

When implementing the algorithms presented in this paper into a software package, one must design stopping criterion for the Lanczos iteration. There are two independent factors affecting the stopping criterion. First, we must be sure that the eigenpair $(\lambda_1^{(k)}, u_1^{(k)})$ of T_k has already converged to the eigenpair (λ_1, w_1) of A, as it is only reasonable to compute the deflated solution of T_k after this has occurred. Secondly, we should stop iterating when $\|x_d - x_d^{(k)}\|$ is reasonably small.

A well-known result in the Lanczos algorithm is that it is not necessary to compute the Ritz vector in order to check for convergence. This is because of the very useful relation [14]

$$Aw_i^{(k)} = \lambda_i^{(k)} w_i^{(k)} + \beta_{k+1} e_k^T u_i^{(k)}, \tag{5.1}$$

from which one derives the residual norm

$$\|(A - \lambda_i^{(k)} I) w_i^{(k)}\| = \beta_{k+1} |e_k^T u_i^{(k)}|. \tag{5.2}$$

In other words the residual norm of the Ritz pair $\lambda_i^{(k)}, w_i^{(k)}$ is equal to the absolute value of the last component of the eigenvector $u_i^{(k)}$ of the tridiagonal matrix multiplied by β_{k+1}. This provides an inexpensive way of checking the convergence of the eigenpair since the error on the eigenvalue is of the order of the square of the residual norm in (5.2) while the angle between the exact eigenvector and the approximate one is of the same order as the residual norm [14].

A formula similar to (5.2) can be easily derived for the residual of the approximate solution $x_d^{(k)}$. In fact we must first define what might be a suitable analogue to the residual norm in the context of nearly singular systems. Ideally, we would like to consider the residual norm for the matrix $PA = (I - w_1 w_1^T)A$, i.e., we would wish to consider $P(b - Ax_d)$, the deflated residual, as an appropriate analogue of the usual residual vector. The reason why this is the correct analogue of the residual norm in the non-sigular case is that, with respect to x_d, we are in fact attempting to solve the system $PAx_d = Pb$ in the subspace PR^n. Unfortunately, the exact projector P depends on the exact eigenvector w_1 which is not known a priori. However, once the approximate eigenvector $w_1^{(k)}$ has converged, we can use it in place of w_1 and therefore define the approximate projected residual as

$$r_d^{(k)} \equiv P^{(k)}(b - Ax_d^{(k)}) \equiv \left(I - w_1^{(k)} \left[w_1^{(k)}\right]^T\right)(b - Ax_d^{(k)}), \tag{5.3}$$

where $P^{(k)}$ is the projector in the direction orthogonal to $w_1^{(k)}$. Substituting equation (4.5) in the above equation we get

$$r_d^{(k)} = P^{(k)}\left(b - A\left[x^{(0)} + V_k z^{(k)}\right]\right) = P^{(k)}\left(r^{(0)} - AV_k z_d^{(k)}\right).$$

Using the relation (3.7) and recalling that $v_1 = r_0/\|r_0\|$, we obtain

$$r_d^{(k)} = P^{(k)}\left(\|r_0\|v_1 - AV_kz_d^{(k)}\right) = P^{(k)}\left(\|r_0\|v_1 - V_kT_kz_d^{(k)} - \beta_{k+1}e_k^Tz_d^{(k)}v_{k+1}\right)$$

$$= P^{(k)}V_k\left(\|r_0\|e_1 - T_kz_d^{(k)}\right) - \beta_{k+1}e_k^Tz_d^{(k)}v_{k+1}$$

Now observe that by (4.1) and (4.2) the term $\|r_0\|e_1 - T_kz_d^{(k)}$ is equal to a scalar multiple of $u_1^{(k)}$, and since $V_ku_1^{(k)} = w_1^{(k)}$ the first term in the above sum vanishes. Hence we have proved that the residual norm is given by

$$\|r_d^{(k)}\| = \beta_{k+1}|e_k^Tz_d^{(k)}|, \tag{5.4}$$

which can be computed at little cost at each iteration.

Peculiar situations may arise in which the convergence of $x_d^{(k)}$, as measured by the above residual norm, occurs before the eigenpair of T_k has converged. Should this happen, the corresponding solution $x_d^{(k)}$ must not be accepted. Accordingly, a general strategy might proceed as follows. We iterate with the Lanczos process, without computing $z_d^{(k)}$ or $x_d^{(k)}$, until the estimate (5.2) indicates that the eigenpair $(\lambda_1^{(k)}, w_1^{(k)})$ has converged. The eigenvector $u_1^{(k)}$ that is used in (5.2) can be computed every few iterations by inverse iteration. After this eigenpair has converged, we performs selective orthogonalization with respect to this Ritz pair in order to prevent it from reappearing in T_k. At the same time, we start computing $x_d^{(k)}$, either explicitly by one of the two deflation algorithms outlined in Section 2, or by an updating procedure similar to one used in SYMMLQ [12]. When the estimate (5.4) indicates that $x_d^{(k)}$ has converged, we stop the Lanczos iteration.

6. Theoretical error bounds

In this section we address the issue of convergence rates and will derive theoretical error bounds on the approximate deflated solution $x_d^{(k)}$. For this purpose we need some additional notation. Recall that Π_k is the orthogonal projector onto the Krylov subspace K_k. We denote by $Q^{(k)} \equiv P^{(k)}\Pi_k$ the orthognal projector onto the subspace $\hat{K}_k$ of K_k, which is orthogonal to the Ritz vector $w_1^{(k)}$. Thus, the rank of Π_k is k in general while the rank of $Q^{(k)}$, i.e., the dimension of the subspace $\hat{K}_k = Q^{(k)}\mathbf{R}^N = P^{(k)}K_k$, is $k-1$ in general. Without loss of generality it is assumed throughout this section that $x^{(0)} = 0$, i.e., $r^{(0)} = b$. We will establish our result with the help of a few simple lemmas.

Lemma 6.1. *The Ritz deflated solution $x_d^{(k)}$ is the (unique) solution of the Galerkin problem*

$$Q^{(k)}(b - Ax) = 0, \quad x \in Q^{(k)}\mathbf{R}^N, \tag{6.1}$$

Proof. From the comments following (4.4), (4.5), it is clear that $x_d^{(k)}$ belongs to the subspace $Q^{(k)}\mathbf{R}^N$. We derive from the Lanczos relation (3.7) that

$$Q^{(k)}(b - Ax_d^{(k)}) = Q^{(k)}(b - AV_kz_d^{(k)}) = Q^{(k)}\left(\|b\|v_1 - V_kT_kz_d^{(k)} - \beta_{k+1}v_{k+1}e_k^Tz_d^{(k)}\right)$$

The term $Q^{(k)}v_{k+1}$ vanishes because v_{k+1} which is orthogonal to the subspace K_k, is also orthogonal to the subspace $Q^{(k)}\mathbf{R}^N$ of K_k. We are left with

$$Q^{(k)}(b - Ax_d^{(k)}) = Q^{(k)}\left(Q^{(k)}\|b\|v_1 - V_kT_kz_d^{(k)}\right)$$

Noticing that $Q^{(k)}v_1 = v_1 - v_1^T w_1^{(k)} w_1^{(k)} = V_k(e_1 - e_1^T u_1^{(k)} u_1^{(k)})$ we finally obtain

$$Q^{(k)}(b - Ax_d^{(k)}) = Q^{(k)}V_k \left(\|b\|(e_1 - e_1^T u_1^{(k)} u_1^{(k)}) - T_k z_d^{(k)} \right)$$

By definition of the deflated solution $z_d^{(k)}$, the above vector is zero.

∎

We now assume that A is positive definite and denote by $\|.\|_A$ the A-norm, i.e., $\|x\|_A = (Ax, x)^{1/2}$. In classical Galerkin techniques, the right hand side b in (6.1) belongs to the subspace of projection so that we approximately solve $Ax = b$ in that subspace. Here, however, the right hand side is not in the subspace of projection. Observing that (6.1) also reads $Q^{(k)}(Q^{(k)}b - Ax) = 0$, we can say that the Lanczos deflation method attempts to solve the linear system

$$Ax = Q^{(k)}b \tag{6.2}$$

by a Galerkin process onto the subspace $\hat{K}_k$. Thus, for the approximation to be accurate, we must show that the projected right hand side $Q^{(k)}b$ is in some sense close to the deflated right hand side Pb. This will be considered in detail later. We now apply a classical argument in Galerkin methods.

In the following discussion we denote by $\hat{x}_k$ the exact solution of the linear system (6.2). As stated above, $x_d^{(k)}$ is the Galerkin solution of the new linear system (6.2), i.e., it is a Galerkin approximation to $\hat{x}_k$ from the subpsace $\hat{K}_k$. A standard theorem relating the Galerkin method to the Rayleigh-Ritz method yields the following result on the error $x_d^{(k)} - \hat{x}_k$.

Lemma 6.2. *The approximate deflated solution $x_d^{(k)}$ minimizes the function*

$$J(x) = \|x - \hat{x}_k\|_A$$

among all elements x of the subspace $\hat{K}_k = Q^{(k)}\mathbf{R}^N$.

We now wish to reformulate the above results in terms of polynomials. Clearly, a vector v is in the Krylov subspace K_k if and only if it can be expressed as $v = p(A)b$ where p is a polynomial of degree $\leq k - 1$. The next lemma shows how the additional constraint that v belongs to $Q^{(k)}\mathbf{R}^N$ translates for the polynomial p.

Lemma 6.3. *A vector v of $\mathbf{R}^N$ belongs to $Q^{(k)}\mathbf{R}^N$ if and only if it is of the form $v = q(A)b$ where q is a polynomial of degree $\leq k - 1$ such that $q(\lambda_1^{(k)}) = 0$.*

Proof. Consider a vector of K_k of the form $v = q(A)v_1$, where q is of degree $\leq k - 1$. This vector is in $\hat{K}_k$ if and only if $(v, w_1^{(k)}) = 0$, i.e., if and only if

$$(q(A)v_1, w_1^{(k)}) = 0. \tag{6.3}$$

The section of the linear operator A in K_k, i.e., the rank k linear operator $A_k = \Pi_k A_{|K_k}$ approximates A in the subspace K_k in the Galerkin process: its matrix representation in the basis V_k is $V_k T_k V_k^T$. It is easy to show that $A_k^j v_1 = \Pi_k A^j v_1$ for any $j, j \leq k$ and therefore we have $q(A)v_1 = \Pi_k q(A_k)v_1$, which substituted in (6.3) yields

$$0 = (\Pi_k q(A_k)v_1, w_1^{(k)}) = (v_1, q(A_k)\Pi_k w_1^{(k)}) = (v_1, q(A_k)w_1^{(k)}). \tag{6.4}$$

Clearly, $w_1^{(k)}$ is an eigenvector of A_k associated with the eigenvalue $\lambda_1^{(k)}$ and as a result the above relation becomes

$$(v_1, q(\lambda_1^{(k)})w_1^{(k)}) = q(\lambda_1^{(k)})(v_1, w_1^{(k)}) = 0 \ .$$

Notice that the inner product $(v_1, w_1^{(k)})$ is the first component of the eigenvector $u_1^{(k)}$ of the tridiagonal matrix, associated with the eigenvalue $\lambda_1^{(k)}$. This component cannot be zero for a nonreducible tridiagonal matrix, so we conclude that v belongs to K_k if and only if $q(\lambda_1^{(k)}) = 0$.

▮

Denoting by $\hat{\mathbf{P}}_\mathbf{k}$ the space of all polynomials q of degree $\leq k$, such that $q(\lambda_1^{(k)}) = 0$, we can state an immediate corollary of the above lemmas.

Corollary 6.1. *The approximation $x_d^{(k)}$ is such that*

$$\|x_d^{(k)} - \hat{x}_k\|_A = \min_{q \in \hat{\mathbf{P}}_\mathbf{k}} \|q(A)b - A^{-1}Q^{(k)}b\|_A.$$

In the next proposition we use this equality to estimate the distance between the exact solution $\hat{x}_k$ of (6.2) and its Galerkin approximation $x_d^{(k)}$.

Proposition 6.1. *Assume that k is sufficiently large that $\lambda_2 - \lambda_1^{(k)} \geq \lambda_1$. Then at the k-th step of the deflated Lanczos procedure we have the inequality*

$$\|x_d^{(k)} - \hat{x}_k\|_A \leq \frac{\lambda_1^{(k)} - \lambda_1}{\lambda_1^{(k)}} C_k \|b\|_A + \left(\frac{1}{\sqrt{\lambda_1}} + \frac{\sqrt{\lambda_N}}{\lambda_2} \right) \ sin\Theta(w_1, w_1^{(k)})\|b\| + \frac{1 + k^2(1+\nu_k)}{T_{k-1}(\nu_k)} \|x_d\|_A \ , \tag{6.5}$$

where T_{k-1} is the Chebyshev polynomial of degree $k-1$ of the first kind,

$$\nu_k = \frac{\lambda_N + \lambda_2 - \lambda_1^{(k)}}{\lambda_N - \lambda_2 + \lambda_1^{(k)}},$$

C_k/k converges to $\frac{2(1+\nu_k)}{\lambda_N\sqrt{\nu^2-1}}$ in which $\nu = lim \ \nu_k$, and $\Theta(x,y)$ represents the acute angle between the vectors x and y in $\mathbf{R}^N$.

Proof. Recall that P denotes the (eigen)-projector onto the subspace orthogonal to w_1, i.e., $P = I - w_1 w_1^T$. For any polynomial q in $\hat{\mathbf{P}}_\mathbf{k}$ we have

$$\begin{aligned} \|x_d^{(k)} - \hat{x}_k\|_A &\leq \|q(A)b - A^{-1}Q^{(k)}b\|_A \leq \|q(A)b - A^{-1}Pb\|_A + \|A^{-1}(P - Q^{(k)})b\|_A \\ &\leq \|q(A)b - A^{-1}Pb\|_A + \|A^{-1}P(P - Q^{(k)})b\|_A + \|A^{-1}(I-P)(P - Q^{(k)})b\|_A \end{aligned}$$

or,

$$\|x_d^{(k)} - \hat{x}_k\|_A \leq \|q(A)b - A^{-1}Pb\|_A + \|A^{-1}P(I - Q^{(k)})b\|_A + \|A^{-1}(I-P)Q^{(k)}b\|_A. \tag{6.6}$$

Consider first the last two terms of the right hand side. Since $Q^{(k)}b = P^{(k)}b$, by using elementary properties of projectors, we get

$$\|A^{-1}P(I - Q^{(k)})b\|_A = \|A^{-1}PP(I - P^{(k)})b\|_A \leq \|A^{-1}P\|_A \|P(I - P^{(k)})b\|_A$$

It is easy to show that

$$\|A^{-1}P\|_A = \|PA^{-1}\|_A = \frac{1}{\lambda_2}. \tag{6.7}$$

Moreover,

$$\|P(I-P^{(k)})b\|_A \leq \sqrt{\lambda_N}\|P(I-P^{(k)})b\| \leq \sqrt{\lambda_N}\|P(I-P^{(k)})\|\|b\|,$$

and

$$\|P(I-P^{(k)})\| = sin\Theta(w_1, w_1^{(k)}). \tag{6.8}$$

The last term in the right hand side of (6.6) satisfies

$$\|A^{-1}(I-P)Q^{(k)}b\|_A = \|A^{-1}(I-P)(I-P)P^{(k)}b\|_A \leq \|A^{-1}(I-P)\|_A\|(I-P)P^{(k)}b\|_A,$$

with $\|A^{-1}(I-P)\|_A = 1/\lambda_1$. The vector $(I-P)Q^{(k)}b$, when expanded with respect to the eigenvectors of A, has only one term corresponding to the first eigenvector. Hence,

$$\|(I-P)P^{(k)}b\|_A = \sqrt{\lambda_1}\|(I-P)P^{(k)}b\| \leq \sqrt{\lambda_1}\|(I-P)P^{(k)}\|\|b\|,$$

and it is again possible to show that

$$\|(I-P)P^{(k)}\| = sin\Theta(w_1, w_1^{(k)}). \tag{6.9}$$

Using these upper bounds for the last two terms of (6.6) we arrive at

$$\|x_d^{(k)} - \hat{x}_k\|_A \leq \|q(A)b - A^{-1}Pb\|_A + \left(\frac{1}{\sqrt{\lambda_1}} + \frac{\sqrt{\lambda_N}}{\lambda_2}\right)\ sin\Theta(w_1, w_1^{(k)})\|b\|. \tag{6.10}$$

We now seek a particular polynomial q in $\hat{\mathbf{P}}_{\mathbf{k}}$ for which the first term in the above expression is as small as possible. Consider the particular polynomial of degree k defined by

$$p(\lambda) = \frac{1}{\lambda_1^{(k)}}\left[\lambda t(\lambda - \lambda_1^{(k)}) - (\lambda - \lambda_1^{(k)})t(\lambda)\right],$$

where $t(\lambda)$ is the polynomial of degree $k-1$ defined by

$$t(\lambda) = \frac{T_{k-1}(\nu_k - \alpha_k\lambda)}{T_{k-1}(\nu_k)}$$

in which

$$\nu_k = \frac{\lambda_N + \lambda_2 - \lambda_1^{(k)}}{\lambda_N - \lambda_2 + \lambda_1^{(k)}}, \qquad \alpha_k = \frac{1+\nu_k}{\lambda_N} = \frac{2}{\lambda_N - \lambda_2 + \lambda_1^{(k)}}.$$

The polynomial $t(\lambda)$ is of degree $k-1$ and its value at the origin is 1. In fact, it is chosen so as to minimize the infinity norm in the interval $[\lambda_2 - \lambda_1^{(k)}, \lambda_N]$, over all such polynomials. We observe that $p(\lambda_1^{(k)}) = p(0) = 1$ which means that p can be written as $p(\lambda) = 1 - \lambda q(\lambda)$. Moreover, q is of degree $k-1$ and we have $q(\lambda_1^{(k)}) = 0$ so that the polynomial q is in $\hat{K}_k$.

Let us expand the vector $A^{-1}b$ in the eigenbasis of A as $A^{-1}b = \sum_{i=1}^N \gamma_i w_i$. Noticing that $PA^{-1}b = A^{-1}b - \gamma_1 w_1$ we get

$$\|q(A)b - A^{-1}Pb\|_A^2 = \|q(A)AA^{-1}b - PA^{-1}b\|_A^2 = \lambda_1[1-p(\lambda_1)]^2\gamma_1^2 + \sum_{i=2}^N \lambda_i\gamma_i^2 p(\lambda_i)^2. \tag{6.11}$$

We start by focussing on the first term of the right hand side in the above equality. The factor $p(\lambda_1) - 1$ satisfies the relations:

$$p(\lambda_1) - 1 = \frac{\lambda_1}{\lambda_1^{(k)}} t(\lambda_1 - \lambda_1^{(k)}) - \frac{\lambda_1 - \lambda_1^{(k)}}{\lambda_1^{(k)}} t(\lambda_1) - 1$$
$$= \frac{\lambda_1}{\lambda_1^{(k)}} [t(\lambda_1 - \lambda_1^{(k)}) - 1] + \frac{\lambda_1^{(k)} - \lambda_1}{\lambda_1^{(k)}} (t(\lambda_1) - 1).$$

Remembering that $t(0) = 1$ this can be rewritten as

$$\frac{p(\lambda_1) - 1}{\lambda_1} = \frac{\lambda_1 - \lambda_1^{(k)}}{\lambda_1^{(k)}} \left[\frac{t(\lambda_1 - \lambda_1^{(k)}) - t(0)}{\lambda_1 - \lambda_1^{(k)}} \right] + \frac{\lambda_1^{(k)} - \lambda_1}{\lambda_1^{(k)}} \left[\frac{t(\lambda_1) - t(0)}{\lambda_1} \right]$$

or,

$$\frac{p(\lambda_1) - 1}{\lambda_1} = \frac{\lambda_1 - \lambda_1^{(k)}}{\lambda_1^{(k)}} \left[t'(\xi_1) - t'(\xi_2) \right],$$

where $\lambda_1 - \lambda_1^{(k)} \leq \xi_1 \leq 0$, $0 \leq \xi_2 \leq \lambda_1$. The derivative $t'(\lambda)$ is given by

$$t'(\lambda) = -(k-1)\alpha_k \frac{U_{k-2}(\nu_k - \alpha_k \lambda)}{T_{k-1}(\nu_k)} \tag{6.12}$$

where U_{k-2} is the Chebyshev polynomial of degree $k-2$ of the second kind. Since $\lambda_1 - \lambda_1^{(k)}$ converges to zero from the left, and since the function $|U_{k-2}(\nu_k - \alpha_k \lambda)|$ decreases in the interval $[\lambda_1 - \lambda_1^{(k)}, \lambda_1]$, we have $|t'(\xi_2)| \leq |t'(\xi_1)| \leq |t'(\lambda_1 - \lambda_1^{(k)})| \equiv C_k$. It is clear that at the limit C_k is equivalent to

$$|t'(0)| = (k-1)\alpha_k \frac{U_{k-2}(\nu_k)}{T_{k-1}(\nu_k)}$$

which in turn can be shown to be asymptotic with

$$\frac{k\alpha_k}{\sqrt{\nu_k^2 - 1}}.$$

Finally, going back to the first term in (6.11),

$$\lambda_1[1 - p(\lambda_1)]^2 \gamma_1^2 = \lambda_1 [\frac{1 - p(\lambda_1)}{\lambda_1}]^2 \lambda_1^2 \gamma_1^2 \leq [\frac{1 - p(\lambda_1)}{\lambda_1}]^2 \|b\|_A^2 \leq \left[2C_k \frac{\lambda_1^{(k)} - \lambda_1}{\lambda_1^{(k)}} \|b\|_A \right]^2 .$$

Using the inequality $(a^2 + b^2)^{1/2} \leq |a| + |b|$ in (6.11) and the above bound we obtain

$$\|q(A)b - A^{-1}Pb\|_A \leq 2C_k \frac{\lambda_1^{(k)} - \lambda_1}{\lambda_1^{(k)}} \|b\|_A + \left[\sum_{i=2}^{N} \lambda_i \gamma_i^2 p(\lambda_i)^2 \right]^{1/2} \tag{6.13}$$

Consider now the second part of the right-hand-side in (6.13). We have

$$p(\lambda_i) = \lambda_i \left[\frac{t(\lambda_i) - t(\lambda_i - \lambda_1^{(k)})}{\lambda_1^{(k)}} \right] + t(\lambda_i - \lambda_1^{(k)}) \tag{6.14}$$

By the mean-value theorem, the expression between brackets can be expressed as $t'(\xi)$, the derivative of $t(\lambda)$ at some point ξ in the interval $[\lambda_i - \lambda_1^{(k)}, \lambda_i]$. The point ξ belongs to the interval $[\lambda_2 - \lambda_1^{(k)}, \lambda_N]$ and therefore its transformed value $(\nu_k - \alpha_k \xi)/\nu_k$ belongs to the interval $[-1, 1]$. From the expression (6.12) and the fact that $|U_{k-2}(x)| \le (k-2)$ for $x \in [-1, 1]$, it is clear that $t'(\xi) \le k^2 \alpha_k / T_{k-1}(\nu_k)$. Moreover, the second term of of the right-hand side of (6.14) is naturally bounded by $1/T_{k-1}(\nu_k)$. Therefore,

$$|p(\lambda_i)| \le \lambda_i \frac{k^2}{T_{k-1}(\nu_k)} + \frac{1}{T_{k-1}(\nu_k)} \le \frac{1 + k^2 \alpha_k \lambda_N}{T_{k-1}(\nu_k)} = \frac{1 + k^2(1+\nu_k)}{T_{k-1}(\nu_k)}. \tag{6.15}$$

Thus, the expression between brackets in (6.13) is bounded from above as follows

$$\left[\sum_{i=2}^{N} \lambda_i \gamma_i^2 p(\lambda_i)^2\right]^{1/2} \le \frac{1 + k^2(1+\nu_k)}{T_{k-1}(\nu_k)} \left[\sum_{i=2}^{N} \lambda_i \gamma_i^2\right]^{1/2} = \frac{1 + k^2(1+\nu_k)}{T_{k-1}(\nu_k)} \|x_d\|_A. \tag{6.16}$$

The result follows by combining (6.16), (6.13), and (6.10).

∎

A few comments on the above proposition are in order. The term *sin* $\Theta(w_1, w_1^{(k)})$ converges to zero like [14]

$$\frac{1}{T_k(\gamma_1)}, \tag{6.17}$$

where

$$\gamma_1 = \frac{\lambda_N + \lambda_2 - \lambda_1}{\lambda_N - \lambda_2}. \tag{6.18}$$

Similarly, the relative error on the eigenvalue $(\lambda_1^{(k)} - \lambda_1)/\lambda_1^{(k)}$ converge to zero decreasingly as the square of the quantity (6.17). Observe the similarity between the number γ_1 and the coefficient ν_k of the proposition. Since $\lambda_1^{(k)}$ is close to zero for large k, these two numbers will be close to each other at the limit. The coefficient C_k in the proposition is not bounded but is of the order of $O(k)$. However, its product with $(\lambda_1^{(k)} - \lambda_1)/\lambda^{(k)}$ tends to zero rapidly. The same observation holds for the term k^2 in the numerator of the last term of (6.5).

We now turn our attention to the actual error $e_k \equiv x_d - x_d^{(k)}$. We cannot use an $A-$norm to measure this error because if λ_1 is small this norm will dampen the component of the error in the w_1 direction. As a result a possible large component in that direction can be unfairly hidden by this norm. Therefore, we choose to use the usual Euclidean norm $\|.\|$. Moreover, we separate the above error in two distinct parts, namely the component $(I-P)e_k$ in the w_1 direction, and the component Pe_k orthogonal to it. We show that both terms tend to zero quickly as k increases.

Proposition 6.2. *The part of the error in the direction of the eigenvector w_1 satisfies the inequality:*

$$\frac{\|(I-P)(x_d - x_d^{(k)})\|}{\|x_d^{(k)}\|} \le sin\ \Theta(w_1, w_1^{(k)}) \tag{6.19}$$

Proof. We have

$$\|(I-P)(x_d - x_d^{(k)})\| = \|(I-P)x_d^{(k)}\| = \|(I-P)P^{(k)}x_d^{(k)}\| \le \|(I-P)P^{(k)}\|\|x_d^{(k)}\|$$

The result follows from (6.9).

∎

The above result means that relatively to $\|x_d^{(k)}\|$, the error in the direction of w_1 is bounded by the sine of the angle between the exact eigenvector and the Ritz vector. As we mentioned above, this angle is known to converge to zero as rapidly as the sequence (6.17) see [14].

Proposition 6.3. *The part of the error in the eigenspace orthogonal to w_1 satisfies the inequality:*

$$\|P(x_d - x_d^{(k)})\| \le \frac{1}{\lambda_2} sin\ \Theta(w_1, w_1^{(k)})\|b\| + \frac{\epsilon_k}{\sqrt{\lambda_2}}, \tag{6.20}$$

where ϵ_k is the right hand side of the inequality (6.5).

Proof. By the triangle inequality,

$$\|P(x_d - x_d^{(k)})\| \le \|P(x_d - \hat{x}_k)\| + \|P(x_d^{(k)} - \hat{x}_k)\|.$$

Since $A^{-1}P = PA^{-1} = PA^{-1}P$, the first term of the right hand side is such that

$$\|P(x_d - \hat{x}_k)\| \equiv \|P(A^{-1}Pb - A^{-1}P^{(k)}b)\| = \|PA^{-1}(Pb - P^{(k)}b)\|$$

$$= \|A^{-1}PP(I - P^{(k)})b\| \le \|A^{-1}P\|\|P(I - P^{(k)}b)\|\|b\| \le \frac{1}{\lambda_2} sin\ \Theta(w_1, w_1^{(k)})\|b\|.$$

For the second term, using the inequality

$$\|Py\| \le \frac{1}{\sqrt{\lambda_2}}\|Py\|_A,$$

we get immediatly that

$$\|P(x_d^{(k)} - \hat{x}_k)\| \le \frac{1}{\sqrt{\lambda_2}}\|P(x_d^{(k)} - \hat{x}_k)\|_A \le \frac{1}{\sqrt{\lambda_2}}\|x_d^{(k)} - \hat{x}_k\|_A \le \frac{\epsilon_k}{\sqrt{\lambda_2}}.$$

∎

7. Numerical Experiments

We now present the results of some numerical experiments to verify the accuracy and stability of the Lanczos-deflation algorithms. All computations were performed on a VAX-780 in single precision, with a relative machine precision of about 10^{-7}.

In these experiments, we employ a simple stopping criterion. The convergence of the eigenpair $(\lambda_1^{(k)}, u_1^{(k)})$ is checked every five Lanczos iterations. The tolerance in the stopping criterion is set to 10^{-5}. When this pair has converged, we stop the algorithm. No further test on the convergence of $x_d^{(k)}$ is made. In fact, in all of our tests the convergence of $x_d^{(k)}$ occurs almost simultaneously.

In a first test we solve the linear system $A_1 x = b$ where $A_1 = diag(10^{-I}, 2, 3, \cdots, n)$, with I varying from 1 to 8; $b^T = (1, \cdots, 1)$, and $n = 100$. The deflated solution to the above problem is $x_d = (0, 1/2, \cdot\cdot, 1/i, \cdot\cdot, 1/n)$. We computed the deflated solution using the Lanczos–Projection, and Lanczos–QL methods. For comparison, we also used a standard conjugate gradient method to solve the test problem and then deflated the solution as follows :

$$x_d = x - (x^T u_1)u_1, \qquad \text{where} \qquad u_1^T = (1, 0, \cdots, 0).$$

Figure 1 shows the relative error in x_d versus I for the three methods. Figure 3 illustrates the simultaneous convergence of the approximate eigenpair and deflated solution.

Our second test repeats the above experiments with the matrix $A_2 = T - (\lambda_1 - \sigma)I$, where $T = Tridiag\{-1, 2, -1\}$, λ_1 is the smallest eigenvalue of T and $\sigma = 10^{-I}$, with I again varying from 1 to 8 and $n = 20$. Note that the smallest eigenvalue of A_2 is σ when σ is small. The solution is chosen such that $x = x_d + w_1$ where x_d is such $(x_d, w_1) = 0$. The right hand side b is then obtained by forming $Ax_d + Aw_1$. Figure 2 shows the relative error in x_d versus I for the three methods. In Figure 3 we illustrate the simultaneous convergence of the eigenpair and of the deflated solution for the first test matrix A_1 with $\sigma = 10^{-4}$. The plot shows the residual norms of the eigenpair and of the deflated solution as given by (5.2) and (5.4) respectively. Figure 4 is a similar illustration of this simultaneous convergence for the matrix A_2.

The numerical results show that both Lanczos deflation methods compute the deflated solution with accuracy close to machine precision independent of the singularity of the matrix A. On the other hand, the conjugate gradient method without deflation becomes unstable as A becomes more singular, especially in the second example. The accuracies displayed by the two deflation methods are very similar.

8. Concluding Remarks

We have presented two different ways of extending the Lanczos algorithm to solving nearly singular systems. Both methods retain the advantages of the classical Lanczos-Conjugate gradient procedure in that they access the matrix A only in the form of matrix by vector products. The overhead of the algorithm over regular Lanczos is limited to deflating a tridiagonal matrix and is negligible compared to the cost of the overall computation. There doesn't seem to be much difference in the performance of the two deflation techniques for tridiagonal matrices. While the QL method is slightly more complicated than the orthogonal projection method, it should be more robust and more easily extensible to higher dimensional nullity problems.

Although we have presented an algorithm which requires the storage of the Lanczos vectors, we should emphasize that an updating version similar to SYMMLQ could easily be derived. Moreover, the techniques developed here can in principle be extended to handle higher dimensional null spaces and nonsymmetric problems.

Acknowledgement: The authors thank Mr. Herbert Beke for his assistance throughout this work.

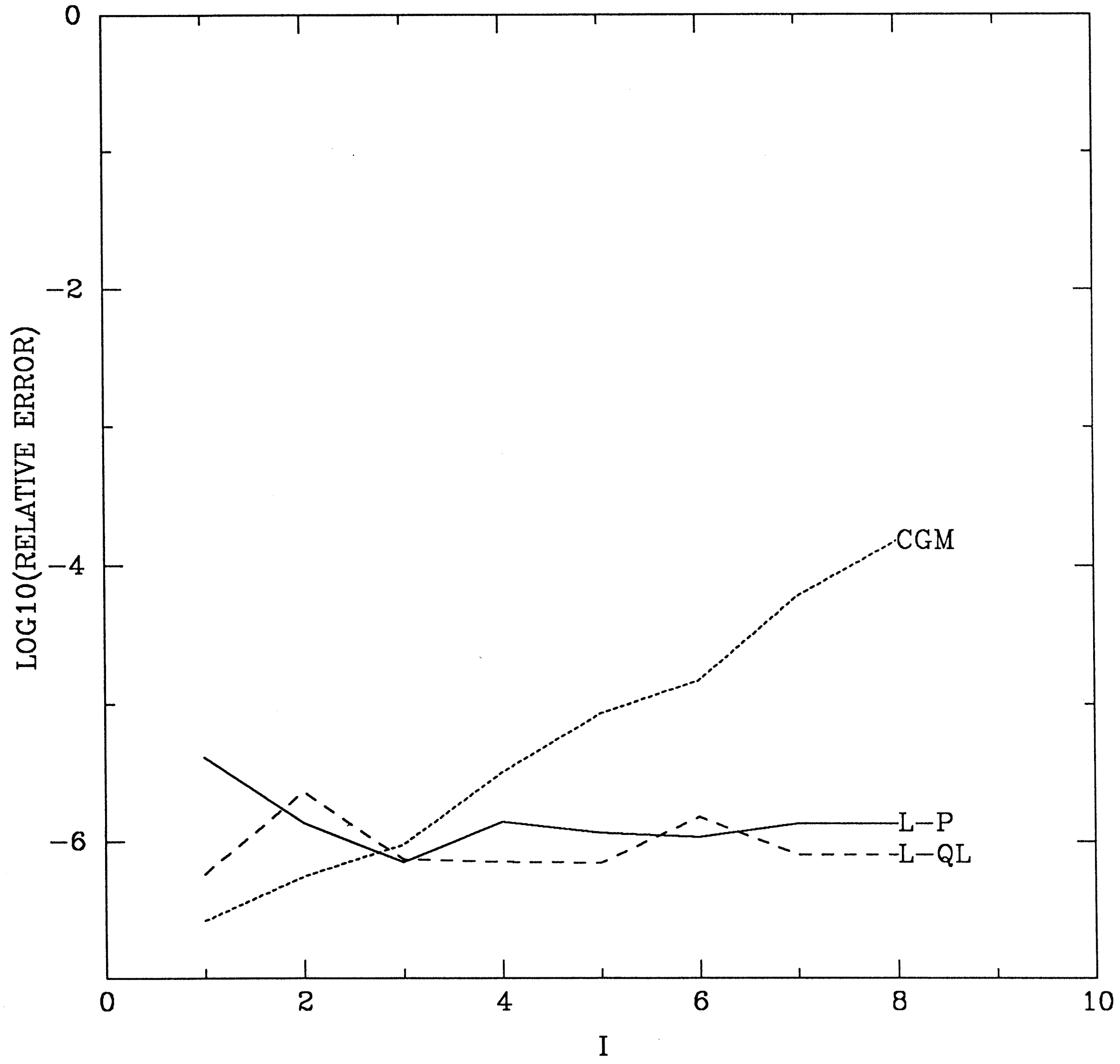

Figure 1: Relative Error versus degree of singularity of three methods: Regular CG (CGM), Lanczos-Projection (L-P) and Lanczos-QL (L-QL).

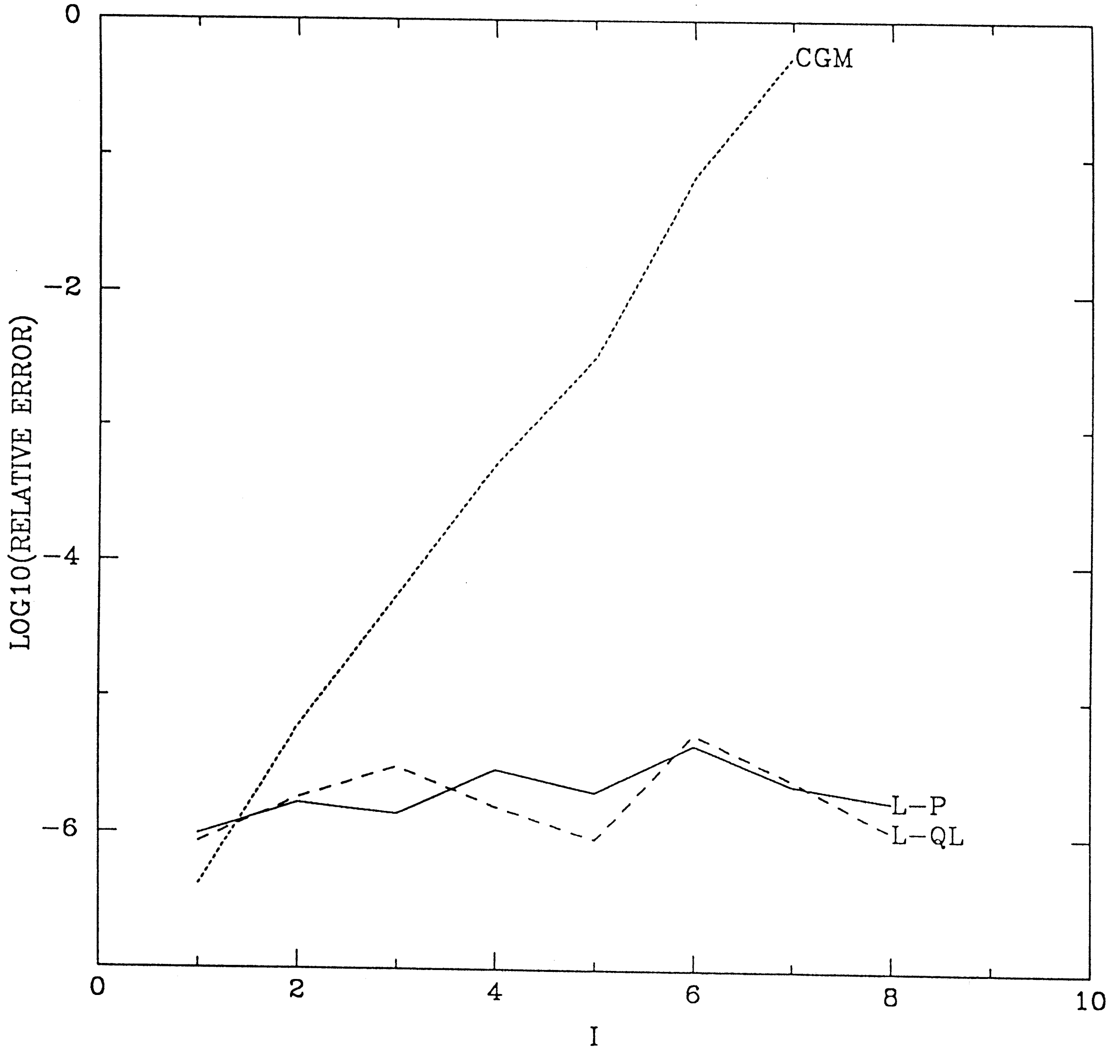

Figure 2: Relative Error versus degree of singularity of three methods: Regular CG (CGM), Lanczos-Projection (L-P) and Lanczos=QL (L-QL).

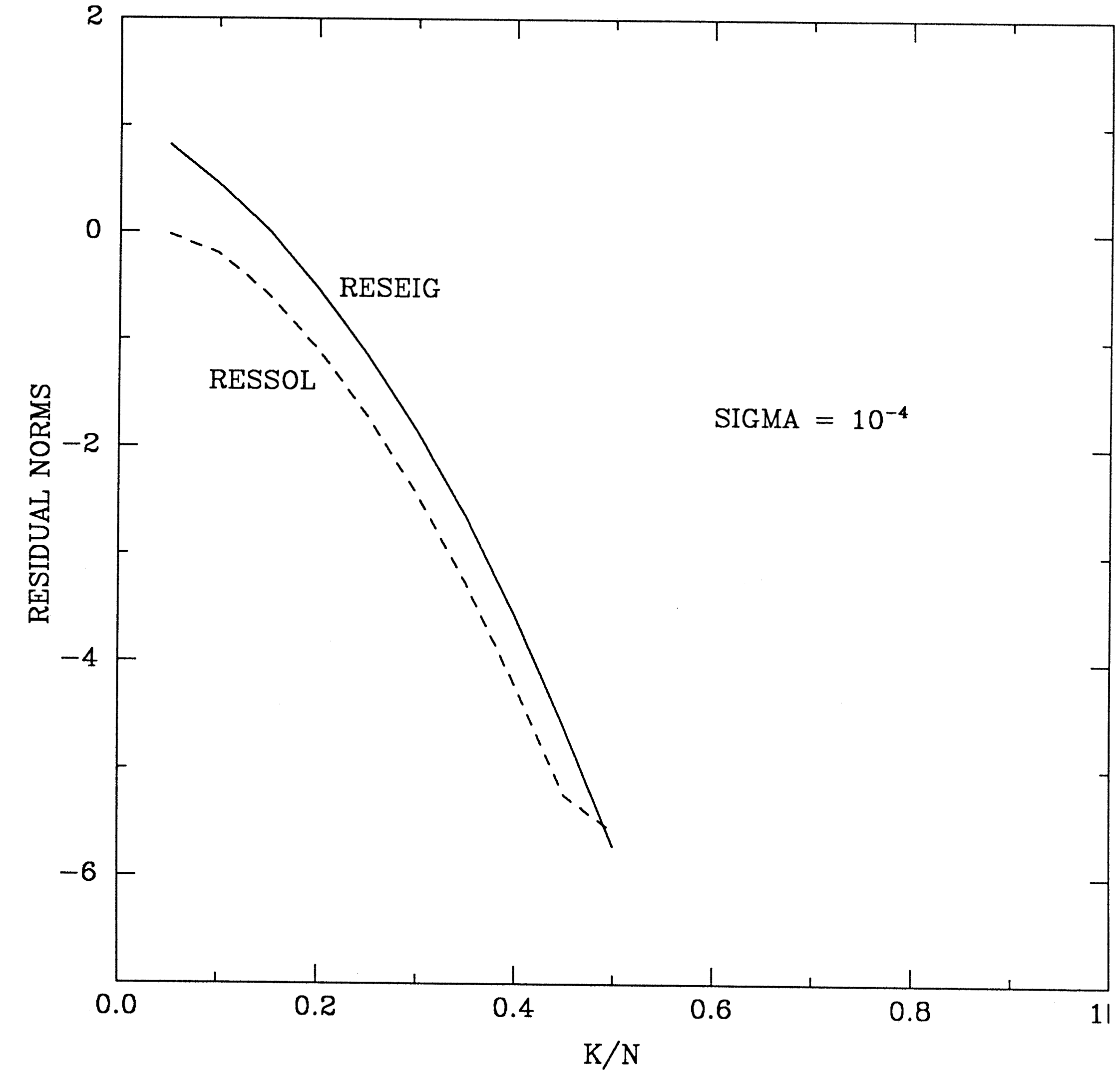

Figure 3: Convergence of eigenpair and deflated solution for A_1.

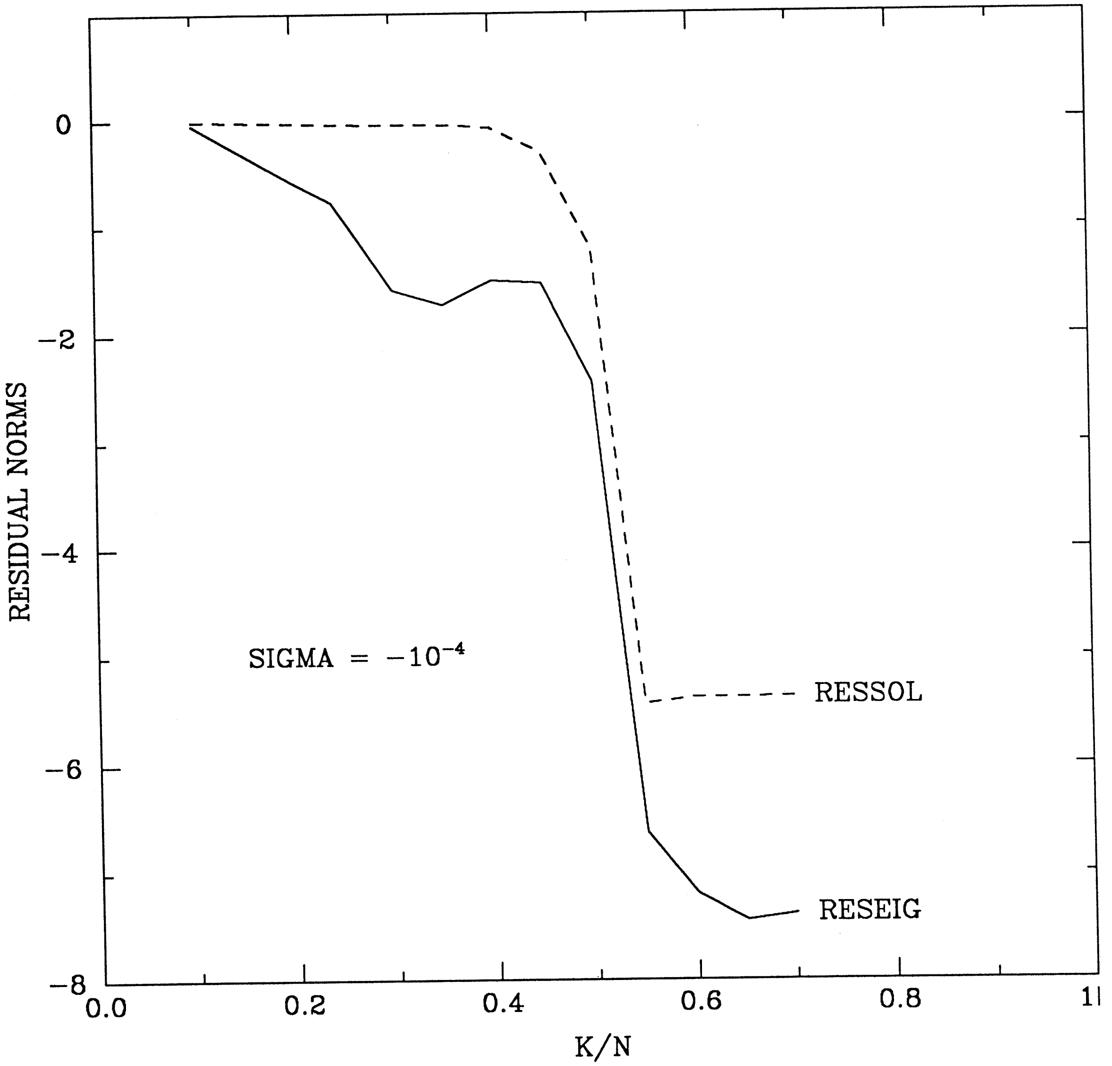

Figure 4: Convergence of eigenpair and deflated solution for A_2.

References

[1] T.F. Chan and Y. Saad, *Iterative Methods for Solving Bordered Systems with Applications to Continuation Methods,* Siam J. Sci. Stat. Comp., 6 (1985), pp. 438–451.

[2] T.F. Chan, *Deflated Decomposition of Solutions of Nearly Singular Systems,* Siam J. Numer. Anal., 1984, 21/4 August (1984), pp. 738–754.

[3] ————, Techniques for Large Sparse Systems Arising from Continuation Methods, T. Kupper, H. Mittelmann and H. Weber eds., *Numerical Methods for Bifurcation Problems,* International Series of Numerical Math., Vol. 70, Birkhauser Verlag, Basel, 1984, pp. 116–128.

[4] T.F. Chan and D. Resasco, *Generalized Deflated Block-Elimination,* Technical Report YALEU/DC/TR-337, Dept. of Computer Science, Yale Univ., 1985.

[5] T.F. Chan, *Deflation Techniques and Block-Elimination Algorithms for Solving Bordered Singular Systems,* Siam J. Sci. Stat. Comp., 5/1 March (1984).

[6] R. Chandra, *Conjugate Gradient Methods for Partial Differential Equations,* Ph.D. Thesis, Yale University, Computer Science Dept., 1978.

[7] P.E. Gill, W. Murray and M. Wright, *Practical Optimization,* Academic Press, New York, 1981.

[8] A. Jepson and A. Spence, Singular Points and Their Computations, T. Kupper, H. Mittelmann and H. Weber eds., *Numerical Methods for Bifurcation Problems,* International Series of Numerical Math., Vol. 70, Birkhauser Verlag, Basel, 1984, pp. 195–209.

[9] H.B. Keller, Numerical Solution of Bifurcation and Nonlinear Eigenvalue Problems, P. Rabinowitz ed., *Applications of Bifurcation Theory,* Academic Press, New York, 1977, pages 359–384.

[10] J.G. Lewis and R.G. Rehm, *The numerical solution of a nonseparable elliptic partial differential equations by preconditioned conjugate gradients,* N.B.S., 85 (1980), pp. 367–389.

[11] J.G. Lewis, *Algorithms for Sparse Matrix Eigenvalue Problems,* Ph.D. Thesis, Stanford University, 1977.

[12] C.C. Paige and M.A. Saunders, *Solution of sparse indefinite systems of linear equations,* SIAM J Numer. Anal., 12 (1975), pp. 617–624.

[13] B.N.Parlett, *A new look a the Lanczos algorithm for solving symmetric systems of linear equations,* Lin. Alg. Appl., 29 (1980), pp. 323–346.

[14] B.N. Parlett, *The Symmetric Eigenvalue Problem,* Prentice Hall, Englewood Cliffs, 1980.

[15] W.C. Rheinboldt, *Numerical Methods for a Class of Finite Dimensional Bifurcation Problems,* SIAM J. of Numer. Anal., 15/1 (1978), pp. 1–11.

[16] Y. Saad, *Krylov subspace methods for solving large unsymmetric linear systems,* Mathematics of Computation, 37 (1981), pp. 105–126.

[17] ————, *Practical use of some Krylov subspace methods for solving indefinite and unsymmetric linear systems,* SIAM J. Sci. Stat. Comp., 5 (1984), pp. 203–228.

[18] D.S. Scott, *Analysis of the symmetric Lanczos process,* Ph.D. Thesis, University of California at Berkeley, 1978.

[19] H. D. Simon, *The Lanczos Algorithm for Solving Symmetric Linear Systems,* Ph.D. Thesis, University of California at Berkeley, 1982.

[20] G.W. Stewart, *On the Implicit Deflation of Nearly Singular Systems of Linear Equations,* SIAM J. Sci. Stat. Comp., 2/2 (1981), pp. 136–140.