Nested Lanczos : Implicitly Restarting a Lanczos Algorithm

Gorik De Samblanx and Adhemar Bultheel

Report TW 267, September 10, 1997

Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

Nested Lanczos : Implicitly Restarting a Lanczos Algorithm

Gorik De Samblanx and Adhemar Bultheel

Report TW267, September 10, 1997

Department of Computer Science, K.U.Leuven

Abstract

In this text, we present a generalisation of the idea of the Implicitly Restarted Arnoldi method to the nonsymmetric Lanczos algorithm, using the two-sided Gram-Schmidt process or using a full Lanczos tridiagonalisation. The Implicitly Restarted Lanczos method can be combined with an implicit filter. It can also be used in case of breakdown and offers an alternative for look-ahead.

AMS Subject Classification. 65F15
keywords: nonsymmetric Lanczos, Implicitly Restarted Arnoldi, two-sided Gram-Schmidt

1 Introduction

In the last few years, the concept of restarting an iterative eigenvalue solver has been widely accepted. Restarting an algorithm – implicitly or explicitly – provides us with a solution for the specific problems that emerge with the use of iterative solvers. These problems can be : slow convergence along with growing computational work per iteration step, the occurrence of spurious solutions or an uncertain robustness.

For Arnoldi's method, the idea of *explicitly* restarting the factorisation was introduced by Saad [13] and different strategies were formulated on how the restart must be applied in practice. The main problem was to find a set of eigenvalues that contains as much 'useful' information as possible. A break-through was accomplished with the *Implicitly* Restarted Arnoldi method (IRA) of Sorensen [14], which lead to the popular ARPACK eigenvalue package [8]. One can say that the advantage of implicitly restarting lies in the fact that it removes uninteresting information instead of trying to retain the interesting part. This approach appeared not only to be cheaper, but it gave a new insight into iterative methods which lead to efficient solutions.

The implicit restarting procedure is normally cheaper than an explicit one, since it does not require to redo the matrix-vector products and the orthogonalisations of the original method. Instead, it performs a QR factorisation of the Hessenberg matrix , i.e. a special Arnoldi run. It can often be combined with an implicit *filtering* step. Restarting the Arnoldi relation can reduce the amount of required storage space for the algorithm, and it can remove lumber information when the method converges too slowly. The filtering can be used in order to remove spurious – but dominant – information that may mislead the algorithm. It also increases the robustness of the algorithm by implementing a validation strategy, or it can be used for acceleration. Many people studied the practical aspects of restarting [9, 15] : how many vectors that should be removed, which vectors contain the most information...

We show in this text how the idea of IRA can be generalised to the non-symmetric Lanczos algorithm. For the symmetric case, Arnoldi corresponds to Lanczos and the theory is much alike [2]. For the non-symmetric case, Grimme et al. [4] considered restarting a *sign symmetric* implementation of the Lanczos algorithm. Our approach is more general in that it can be applied to any Lanczos implementation. Restarting the Lanczos algorithm has an additional application that might be very important. If there is a breakdown in the Lanczos algorithm, then the restart can be used in order to remove this singularity.

The aim of this paper is to show how the nonsymmetric Lanczos algorithm can be restarted implicitly in a most general way. However it can be seen as a generalisation of IRA, it contains new and unexpected aspects, e.g. the application in case of breakdown. In Section 2, we briefly recall the nonsymmetric Lanczos algorithm and the different types of breakdown. In Section 3, we restart the Lanczos algorithm by using a bi-orthogonal factorisation of the small, projected problem. This factorisation can be computed by a Lanczos algorithm with special starting vectors or by the two-sided Gram-Schmidt algorithm (BioGS). Both approaches are mathematically equivalent. Section 4 focuses on the 'classical' properties of an implicitly restarted method : the filtering property and the use of 'exact shifts' in order to remove eigenvalues from the approximation. These properties are used to show that the computation of the implicit restart does not break down itself, unless the explicit restart with corresponding starting vectors would break down. In Section 5, we show that implicitly restarted Lanczos may be used as an alternative for look-ahead, in case of breakdown. Look-ahead extends the approximating subspaces until it finds a set that is no longer singular. Implicitly restarted Lanczos choses a fixed number of vectors from this large set, which form a nonsingular pair of subspace bases. In Section 6, we close the text with some conclusions.

Notation : Roman characters denote matrices and vectors; the (i,j)-th entry of a matrix Q is denoted by $q_{i,j}$. The hermitian transpose is given by Q^*. Greek characters denote scalars and $\bar{\alpha}$ is the complex conjugate of α. The k-th unit vector is e_k, the identity matrix is $I_{k,l} \in \mathbf{C}^{k\times l}$ and the column range of a matrix is denoted by $\mathcal{R}(Q)$. Two matrices Q and Z are called bi-orthogonal if $Q^*Z = I$. If they are square, then $Q^*Z = I = QZ^*$.

2 The non-symmetric Lanczos algorithm

Before we show how to restart the Lanczos algorithm, we describe the algorithm itself [7]. The nonsymmetric Lanczos algorithm reduces a matrix to tridiagonal form, however the result can not be guaranteed for an arbitrary matrix. There are many different implementations possible, but all are equivalent upto some scaling of the Lanczos vectors [11, Th. 2.2]. The restarting procedure that we present, does not depend on the exact implementation of the algorithm, so we are free to choose one. It is well known that if the Lanczos algorithm converges, then the bi-orthogonalisation can become errorneous. We assume in the text that this problem is handled, e.g. by using some reorthogonalisation steps.

The Lanczos algorithm projects a complex, nonsymmetric matrix A with an oblique projector. The projector is defined by a bi-orthogonal pair of matrices (V_k, W_k), which represent the basis of the subspaces in which the left and right eigenvectors are approximated. The columns of the basis matrices are called *Lanczos vectors.* The range of the Lanczos vectors only depends on the first columns of $V_k = [v_1, \ldots, v_k]$ and $W_k = [w_1, \ldots, w_k]$. They correspond each to a Krylov subspace

$$\mathcal{R}(V_k) = \mathcal{K}_k(A, v_1) = \{v_1, Av_1, \ldots, A^{k-1}v_1\} \text{ and } \mathcal{R}(W_k) = \mathcal{K}_k(A^*, w_1) = \{w_1, A^*w_1, \ldots, (A^*)^{k-1}w_1\}. \quad (1)$$

The nonsymmetric Lanczos algorithm that we will use is given by Algorithm 1. Unless it breaks down, the algorithm computes a bi-orthogonal pair of basis matrices (V_k, W_k) and an unreduced tridiagonal matrix T_k such that

$$AV_k = V_k T_k + \beta_{k+1} v_{k+1} e_k^* \quad (2)$$

$$A^* W_k = W_k T_k^* + \bar{\gamma}_{k+1} w_{k+1} e_k^* \quad (3)$$

with

$$W_{k+1}^* V_{k+1} = I \quad (4)$$

and

$$T_k = \begin{bmatrix} \alpha_1 & \gamma_2 & & & \\ \beta_2 & \alpha_2 & \gamma_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{k-1} & \alpha_{k-1} & \gamma_k \\ & & & \beta_k & \alpha_k \end{bmatrix}. \quad (5)$$

A tridiagonal matrix is called *unreduced* if all its off-diagonal elements are nonzero. Equations (2) and (3) are called the *Lanczos relations.* In an eigenvalue context, they can be seen as approximate Schur decompositions of A and A^*, from which approximate eigenvalues and eigenvectors can be computed as follows. A pair (θ_i, y_i) is an *approximate (right) eigenpair of A* if

$$y_i \equiv V_k u_i \quad \text{and} \quad T_k u_i - \theta_i u_i = 0, \; i = 1, \ldots, k. \quad (6)$$

We assume that there exist k linear independent vectors y_i. The corresponding (right) residual norm can be computed as

$$||Ay_i - \theta_i y_i||_2 = |u_{i,k}\beta_{k+1}|||v_{k+1}||_2. \quad (7)$$

The algorithm has converged if $|u_{i,k}\beta_{k+1}|||v_{k+1}||_2 \simeq 0$. Analogously, a left eigenpair $(\bar{\theta}_i, y_i^{(*)})$ can be defined by

$$y_i^{(*)} \equiv W_k u_i^{(*)} \quad \text{and} \quad T_k^* u_i^{(*)} - \bar{\theta}_i u_i^{(*)} = 0, \; i = 1, \ldots, k, \quad (8)$$

such that

$$||A^* y_i^{(*)} - \bar{\theta}_i y_i^{(*)}||_2 = |u_{i,k}^{(*)}\bar{\gamma}_{k+1}|||w_{k+1}||_2. \quad (9)$$

The residuals of the approximate eigenpairs fulfil the following Galerkin conditions

$$Ay_i - \theta_i y_i \perp \mathcal{R}(W_k) \quad \text{and} \quad A^* y_i^{(*)} - \bar{\theta}_i y_i^{(*)} \perp \mathcal{R}(V_k), \quad (10)$$

which can be used as a definition for (θ_i, y_i). Other definitions of approximate eigenvalues and eigenvectors are possible, e.g. analogous to the concept of Harmonic Ritz values [10].

Algorithm 1 *Simple Lanczos*

In: $A \in \mathbf{C}^{n\times n}$,
$v_1, w_1 \in \mathbf{C}^n$, *with* $||v_1||_2 = ||w_1||_2 = 1$, $k \in \mathbf{N}$
Out: $V_k, W_k \in \mathbf{C}^{n\times k}$,
$\alpha_i, \beta_i, \gamma_i$ *for* $i = 1, \ldots, k$

1. Set $V_1 = [v_1]$, $W_1 = [w_1]$, $\beta_1 = 0 = \gamma_1$
2. For $j = 1, 2, \ldots, k-1$ *do*
2.1 Set $v_+ = Av_j$ *and* $w_+ = A^* w_j$
2.2 Compute $\alpha_j = w_j^* v_+$

2.3 *Bi-orthogonalise*
$v_+ = v_+ - \alpha_j v_j - \beta_j v_{j-1}$
$w_+ = w_+ - \bar{\alpha}_j w_j - \bar{\gamma}_j w_{j-1}$
2.4 *If* $w_+^* v_+ \neq 0$ *then normalise*
$v_{j+1} = v_+/\beta_{j+1}$
$w_{j+1} = w_+/\bar{\gamma}_{j+1}$
with $\beta_{j+1}\gamma_{j+1} = w_+^* v_+$.
Else Breakdown and Stop.

It is well known that the Lanczos algorithm can suffer from breakdown. Breakdown occurs whenever $w_+^* v_+ = 0$ (Step 2.4 in Alg. 1), such that no β_{k+1} and γ_{k+1} can be found that normalise the new basis vectors v_{k+1} and w_{k+1}. Breakdown can have different causes, which can be classified as follows.

Lucky Breakdown : If $v_+ = 0$ or $w_+ = 0$, then an invariant subspace is found for A or A^*. E.g. if $v_+ = 0$, then the approximate eigenvectors in $\mathcal{R}(V_k)$ are true eigenvectors for the problem – supposing that (2) holds exactly. The approximate eigenvalues θ_i also are equal to eigenvalues of A. If the wanted eigenvalues are among these eigenvalues, then the algorithm may stop. Otherwise, we can set $\beta_{k+1} \equiv 0$ and set v_{k+1} equal to any vector that fulfils the bi-orthogonality condition (4) and proceed with the algorithm. At this point, the matrix T_{k+1} is no more unreduced, since it contains a zero subdiagonal element.

Serious Breakdown : If $v_+ \neq 0$ and $w_+ \neq 0$ but $w_+^* v_+ = 0$, then the Lanczos algorithm can not be continued any more. This event will occur if the *moment matrix* M_k is singular for some k.

Definition 2.1 *Given a matrix* $A \in \mathbf{C}^{n\times n}$ *and two vectors* $v_1,\ w_1 \in \mathbf{C}^n$, *the moment matrix* $M_k(A, v_1, w_1)$ *is defined as*

$$M_k(A, v_1, w_1) = \left[\begin{array}{cccc} w_1 & A^* w_1 & \cdots & (A^*)^{k-1} w_1 \end{array}\right]^* \left[\begin{array}{cccc} v_1 & A v_1 & \cdots & A^{k-1} v_1 \end{array}\right]. \quad (11)$$

A submatrix of a moment matrix is denoted by $M_k^l(A, v_1, w_1) = M_k(A, A^i v_1, (A^*)^j w_1)$, *with* $i + j = l$.

If the parameters in this definition are obvious, then we omit them : $M_k \equiv M_k(A, v_1, w_1)$ and $M_k^l = M_k^l(A, v_1, w_1)$. The matrix $M_k(A, v_1, w_1)$ is a Hankel matrix, since its (i, j)-th element $w_1^* A^{i+j-2} v_1$ is equal to its $(i - m, j + m)$-th element. If serious breakdown occurs, then the algorithm must be continued in a different way. One can restart the method explicitly by chosing new starting vectors v_1 and w_1. But this is expensive and does not guarantee to avoid a new breakdown. A better solution is to give up the tridiagonal form of T_k and perform a Look-Ahead strategy, e.g. [1, 3, 12]. Roughly speaking, a look-ahead strategy tries to find a $p > 0$ such that M_{k+p} has full rank. After the look-ahead step, the algorithm continues, with a block tridiagonal matrix T_{k+p}. We will show that an implicitly restarted Lanczos algorithm offers an alternative for look-ahead.

Incurable breakdown : If none of the $M_k, M_{k+1}, \ldots, M_n$ has full rank, then the breakdown is called *incurable*, since it can not be cured with look-ahead. The only solution then is to restart the algorithm with a new pair of starting vectors. One can prove that if incurable breakdown occurs, then the eigenvalues of T_k are correct eigenvalues of A [16], as with lucky breakdown. However, the approximate eigenvectors are no true eigenvectors.

Summarising, we can say that if breakdown occurs, then the algorithm must be restarted or the matrix T_k can not be longer an unreduced tridiagonal matrix.

3 Implicitly Restarted Lanczos

Let us now show how the Lanczos algorithm can be restarted implicitly. With the term 'implicitly restarting', we denote the reorganisation of equations (2) and (3) such that the new relations are correct Lanczos relations. This means that in exact arithmetic, they could have been generated by the Lanczos algorithm if we restarted it with the proper starting vectors. Since the new matrices V^+ and W^+ must be bi-orthogonal, the reorganisation must consist of a bi-orthogonal transformation. Two algorithms can be used in order to compute this transformation : the Lanczos algorithm itself or the two-sided Gram-Schmidt algorithm [12, p.107].

Theorem 3.1 *Given the Lanczos relations (2) and (3) and suppose that there exist a pair of bi-orthogonal upper Hessenberg matrices* $Q,\ Z \in \mathbf{C}^{k\times k}$ *and a tridiagonal matrix* $H \in \mathbf{C}^{k\times k}$ *such that, for some* $\mu \in \mathbf{C}$,

$$(T_k - \mu I)Q = QH \quad \text{and} \quad (T_k - \mu I)^* Z = ZH^*, \tag{12}$$

then the matrices

$$V_{k-1}^+ \equiv V_k Q I_{k,k-1}, \quad W_{k-1}^+ \equiv W_k Z I_{k,k-1} \quad \text{and} \quad T_{k-1}^+ \equiv I_{k,k-1}^*(H+\mu I)I_{k,k-1} \tag{13}$$

define a restarted Lanczos relation

$$\begin{aligned} AV_{k-1}^+ &= V_{k-1}^+ T_{k-1}^+ + \beta_k^+ v_k^+ e_{k-1}^*, & (14)\\ A^* W_{k-1}^+ &= W_{k-1}^+ (T_{k-1}^+)^* + \bar{\gamma}_k^+ w_k^+ e_{k-1}^*, & (15)\end{aligned}$$

with

$$\begin{aligned} \beta_k^+ v_k^+ &= h_{k,k-1} V_k Q e_k + \beta_{k+1} q_{k,k-1} v_{k+1} & (16)\\ \bar{\gamma}_k^+ w_k^+ &= \bar{h}_{k-1,k} W_k Z e_k + \bar{\gamma}_{k+1} z_{k,k-1} w_{k+1}. & (17)\end{aligned}$$

Proof We prove the theorem only for the first Lanczos relation (2). The proof for (3) is analogous. If we shift (2) with μ and multiply it on the right by Q, then we get (reminding that Q is Hessenberg),

$$\begin{aligned} (A-\mu I)V_k Q &= V_k(T_k - \mu I)Q + \beta_{k+1} v_{k+1} e_k^* Q & (18)\\ &= V_k (T_k - \mu I) Q + \beta_{k+1} v_{k+1} (q_{k,k-1} e_{k-1}^* + q_{k,k} e_k^*). & (19)\end{aligned}$$

If we insert $I = QZ^*$, then we find

$$(A - \mu I)(V_k Q) = (V_k Q)\,(Z^*(T_k - \mu I)Q) + [0 \cdots 0,\ \beta_{k+1} q_{k,k-1} v_{k+1},\ \beta_{k+1} q_{k,k} v_{k+1}], \tag{20}$$

where $Z^*(T_k - \mu I)Q = H$. If we now remove the last column of this equation and shift it back, then we find, using (13),

$$AV_{k-1}^+ = (V_k Q)(H + \mu I) I_{k,k-1} + \beta_{k+1} q_{k,k-1} v_{k+1} e_{k-1}^*. \tag{21}$$

Inserting $I_k = I_{k,k-1} I_{k,k-1}^* + e_k e_k^*$, we finally derive

$$\begin{aligned} AV_{k-1}^+ &= (V_k Q) I_{k,k-1} I_{k,k-1}^* (H+\mu I) I_{k,k-1} + (V_k Q) e_k e_k^* (H + \mu I) I_{k,k-1} + \beta_{k+1} q_{k,k-1} v_{k+1} e_{k-1}^* & (22)\\ &= V_{k-1}^+ T_{k-1}^+ + (h_{k,k-1} V_k Q e_k + \beta_{k+1} q_{k,k-1} v_{k+1}) e_{k-1}^*. & (23)\end{aligned}$$

This corresponds to (14), where $\beta_k^+ v_k^+ = h_{k,k-1} V_k Q e_k + \beta_{k+1} q_{k,k-1} v_{k+1}$. The new basis matrices are bi-orthogonal, since

$$(W_{k-1}^+)^* V_{k-1}^+ = I_{k,k-1}^* Z^* W_k^* V_k Q I_{k,k-1} = I_{k,k-1}^* Z^* Q I_{k,k-1} = I_{k,k-1}^* I_{k,k-1} = I_{k-1,k-1}. \tag{24}$$

Also $(W_{k-1}^+)^* v_k^+ = 0$, unless $\beta_k^+ v_k^+ = 0$, because if we multiply (14) by w_{k-1}^+ and combine this with $(W_{k-1}^+)^* A V_{k-1}^+ = T_{k-1}^+$, then the result follows. Notice that since $v_{k+1} \notin \mathcal{R}(V_k)$, $\beta_k^+ v_k^+ \neq 0$. □

The matrix pair (Q, Z) can be computed by a full run of the Lanczos algorithm applied to the small, tridiagonal matrix T_k. Indeed, if we perform Lanczos on a tridiagonal matrix, using starting vectors that are non-zero only in their first two entries, then the resulting matrices will be upper Hessenberg and bi-orthogonal. Unless the Lanczos process on the tridiagonal matrix T_k breaks down.

Lemma 3.2 *Given a tridiagonal matrix* $T \in \mathbf{C}^{k\times k}$ *and two vectors* $q_1,\ z_1 \in \mathbf{C}$ *such that* $q_1^* z_1 = 1$. *If* q_1 *and* z_1 *can be written as*

$$q_1 = q_{1,1} e_1 + q_{1,2} e_2 \quad \text{and} \quad z_1 = z_{1,1} e_1 + z_{1,2} e_2, \tag{25}$$

then the matrices Q *and* Z *that are generated from* q_1, z_1 *and* T *using a Lanczos algorithm will be upper Hessenberg. Moreover, if* $z_1 \equiv z_{1,1} e_1$, *then* Z *will be upper triangular and* Q *lower bi-diagonal.*

Proof The proof can be easily done by induction. If $q_i \in \mathcal{R}(I_{i+1,i})$ then $Tq_i \in \mathcal{R}(I_{i+2,i+1})$, so each new column of Q will have an additional nonzero entry. Therefore, Q must be upper Hessenberg. The same holds for Z. If $z_1 = z_{1,1} e_1$, then clearly $\mathcal{R}(ZI_{k,i}) = \mathcal{R}(I_{k,i})$, so the strict upper triangular part of Q must be zero in order to fulfil $Z^* Q = I$. □

There is a second possibility to compute Q and Z by using the two-sided Gram-Schmidt algorithm (BioGS) [12, p.107]. In general, BioGS computes two bi-orthogonal basis matrices Q and Z for the column range of a set of matrices $F,\ G \in \mathbf{C}^{k\times l}$, i.e.

$$Z^*Q = I_l, \quad \mathcal{R}(F) = \mathcal{R}(Q) \quad \text{and} \quad \mathcal{R}(G) = \mathcal{R}(Z). \tag{26}$$

The BioGS algorithm can also be seen as an algorithm that computed a 'bi-orthogonal QR-factorisation', since it also produces two upper triangular matrices R_Q and $R_Z \in \mathbf{C}^{l\times l}$ such that

$$F = QR_Q \quad \text{and} \quad G = ZR_Z. \tag{27}$$

If F and G have full rank, then R_Q and R_Z will be invertible. An implementation of the two-sided Gram-Schmidt algorithm is given by Algorithm 2.

Algorithm 2 *BioGS*

In: $F, G \in \mathbf{C}^{k\times l}$, $(k \geq l)$

Out: $Q, Z \in \mathbf{C}^{k\times l}$, $R_Q, R_Z \in \mathbf{C}^{l\times l}$

1. Set $Q = [\,] = Z$ *and* $R_Q = [\,] = R_Z$

2. For $j = 1, \ldots, k$ *do*

2.1 Set $f_j = Fe_j$ *and* $g_j = Ge_j$

2.2 Compute $r_{Q,j} = Z^* f_j$, $q_{Z,j} = Q^* g_j$

2.3 Bi-orthogonalise

$f_j = f_j - Qr_{Q,j}$

$g_j = g_j - Zr_{Z,j}$

2.4 If $g_j^* f_j \neq 0$ *then normalise*

$q_j = f_j/\rho_Q$,

$z_j = g_j/\rho_Z$

with $\bar{\rho}_Z \rho_Q = g_j^* f_j$

Else Breakdown and Stop

2.5 Set

$$R_Q = \begin{bmatrix} R_Q & r_{Q,j} \\ 0 & \rho_Q \end{bmatrix}$$

$$R_Z = \begin{bmatrix} R_Z & r_{Z,j} \\ 0 & \rho_Z \end{bmatrix}$$

The BioGS algorithm can break down, even if $\operatorname{rank}(G^*F) = l$. We assume that for our application, this will not occur. We use the BioGS algorithm with $F = T_k - \mu I$ and $G = T_k^* - \bar{\mu} I$, in order to apply Theorem 3.1.

Lemma 3.3 *Suppose that* $T \in \mathbf{C}^{k\times k}$ *is an unreduced tridiagonal matrix, and* $F = T - \alpha I$ *and* $G = (T - \beta I)^*$ *which are decomposed as in (27) with the BioGS algorithm. If* V_k *and* W_k *are the results of* k *Lanczos steps (without breakdown) applied to* T *with* $v_1 = Fe_1$ *and* $w_1 = Ge_1$*, then there exist a diagonal scaling matrix* D *such that* $Q = V_k D$ *and* $Z^* = D^{-1} W_k^*$.

Proof It is clear that there exist a ξ and a ζ such that $q_1 = \xi(T - \alpha I)e_1 = \xi v_1$ and $z_1 = \zeta(T - \beta I)^* e_1 = \zeta w_1$. Suppose that $TV_k = V_k H$ and $T^* W_k = W_k H^*$. Since $V_k,\ W_k \in \mathbf{C}^{k\times k}$ are unreduced upper Hessenberg (apply Lemma 3.2 on a Lanczos run without breakdown), $[e_1,\ V_{k-1}]$ and $[e_1,\ W_{k-1}]$ are full rank upper triangular. If we combine this with the Lanczos relation $TV_{k-1} = V_k H I_{k,k-1}$, then

$$\begin{aligned} (T - \alpha I)[e_1,\ V_{k-1}] &= V_k[e_1,\ (H - \alpha I)I_{k,k-1}] \\ \Rightarrow F = (T - \alpha I) &= V_k\left([e_1,\ (H - \alpha I)I_{k,k-1}][e_1,\ V_{k-1}]^{-1}\right) = V_k R_V. \end{aligned}$$

Analogously,

$$G = (T - \beta I)^* = W_k\left([e_1,\ (H - \beta I)I_{k,k-1}][e_1,\ W_{k-1}]^{-1}\right) = W_k R_W.$$

Upto scaling of the matrices V_k and W_k, this corresponds to (27). □

The restarting procedure of Theorem 3.1 requires $k-1$ steps of the small Lanczos algorithm, i.e. the Lanczos algorithm applied to the small matrix $T_k - \mu I$. Only the first $k-1$ columns of Q and Z are used, along with the last off-diagonal elements of H, which are also computed at step $k-1$. The following lemma sets a condition for the algorithm that computes Q and Z under which it will not break down.

Lemma 3.4 *Consider Theorem 3.1. If the matrices Q and Z are computed using the Lanczos (or BioGS) algorithm, then this algorithm will not break down unless for some $j < k$, $W_j^*(A-\mu I)^2 V_j$ does not have full rank.*

Proof The algorithm breaks down at step j if $M_j' \equiv ((T_k - \mu I)^* I_{k,j})^* (T_k - \mu I) I_{k,j}$ is singular. Since this matrix is equal to $M_j' = (W_k(T_k - \mu I)^* I_{k,j})^* V_k(T_k - \mu I) I_{k,j}$, we can insert (2) and (3), so

$$M_j' = ((A-\mu I)^* W_k I_{k,j})^* (A-\mu I) V_k I_{k,j} = W_j^* (A-\mu I)^2 V_j .$$

□

The result of Lemma 3.4 can be interpreted in terms of the implicitly restarted process. Since a Krylov subspace invariant when A is shifted, the subspace $\mathcal{R}(V_k) = \mathcal{K}_k(A, v_1) = \mathcal{K}_k(A-\mu I, v_1)$ and a similar relation holds for $\mathcal{R}(W_k)$.

Lemma 3.5 *Given A, v_1, w_1 as in Definition 2.1, then* $\mathrm{rank}(M_k(A, v_1, w_1)) = \mathrm{rank}(M_k(A-\mu I, v_1, w_1))$.

Proof The lemma follows from the fact that $\mathcal{K}_k(A-\mu I, v_1) \subset \mathcal{K}_k(A, v_1)$, since $(A-\mu I)^m v_1 = \sum_{i=0}^{m} \frac{m!}{(m-i)!i!} \mu^{m-i} A^i v_1 \in \mathcal{K}_k(A, v_1)$, and for the same reason $\mathcal{K}_k(A, v_1) \subset \mathcal{K}_k((A-\mu I)+\mu I, v_1) = \mathcal{K}_k(A-\mu I, v_1)$. Analogously, $\mathcal{K}_k(A^*, w_1) = \mathcal{K}_k((A-\mu I)^*, w_1)$. Hence, $\mathrm{rank}(M_k(A, v_1, w_1)) = \mathrm{rank}(M_k(A-\mu I, v_1, w_1))$. □

Therefore, $M_k' = W_k^*(A-\mu I)^2 V_k$ corresponds to the moment matrix $M_k(A, (A-\mu I)v_1, (A-\mu I)w_1) = M_k^2((A-\mu I), v_1, w_1)$ of the Lanczos process with starting vectors $v_1^+ = (A-\mu I)v_1$ and $w_1^+ = (A-\mu I)^* w_1$. Lemma 3.4 shows that the small Lanczos iteration will not break down if the 'large' Lanczos process with (v_1^+, w_1^+) does not break down. In the following section, we show in Theorem 4.1 that this is exactly the *implicit* Lanczos run. So the implicit restart does not break down if the corresponding explicit restart does not break down, which is not that surprising. An analogous result was found by Grimme et al. [4, Th. 3] for the sign-symmetric case.

Algorithm 3 gives an implementation for Implicitly Restarted Lanczos.

Algorithm 3 *Implicitly Restarted Lanczos (IRL)*
In: V_{k+1}, W_{k+1}, T_k, β_{k+1}, γ_{k+1}, μ_Q, μ_Z
Out: V_k^+, W_k^+, T_{k-1}^+, β_k^+, γ_k^+

1. *Compute Q, Z and R_Q, $R_Z \in \mathbf{C}^{k\times k}$ as*
   $T_k - \mu_Q I = Q R_Q$ *and* $(T_k - \mu_Z I)^* = Z R_Z$
2. *Set $V_{k-1}^+ = V_k Q I_{k,k-1}$, $W_{k-1}^+ = W_k Z I_{k,k-1}$ and $T_{k-1}^+ = I_{k-1,k} Z^* T_k Q I_{k,k-1}$.*
3. *Compute* $v_+ = h_{k,k-1} V_k Q e_k + \beta_{k+1} q_{k,k-1} v_{k+1}$ *and*
   $w_+ = \bar{h}_{k-1,k} W_k Z e_k + \bar{\gamma}_{k+1} z_{k,k-1} w_{k+1}$.
4. *Set $v_k^+ = v_+/\beta_k^+$ and $w_k^+ = w_+/\gamma_k^+$ with $\beta_k^+ \bar{\gamma}_k^+ = w_+^* w_+$.*

4 Applying an Implicit Filtering step

The choice of the shift μ in Theorem 3.1 or α and β is Lemma 3.3 is crucial for an efficient restart of the Lanczos algorithm. Indeed, if μ would be chosen such that most of the 'wanted' information is removed from the subspaces V_k and W_k, then the algorithm would slow down, since this information must be recovered. On the other hand, μ can be used in order to remove uninteresting information, e.g. eigenvectors that correspond to unwanted eigenvalues. As for IRA [14], we prove two properties for the restart. First, we show that an implicit restart that is performed with the BioGS algorithm, implicitly applies a polynomial filter on V_k and W_k, as if it performed a step of subspace iteration. Then we show that the concept of *exact shifts* (a notation that was first introduced for IRA) also holds for the Lanczos case : if μ is equal to an approximate eigenvalue, then this approximate eigenvalue will be removed from the spectrum of T_k. The other eigenvalues remain unaltered.

Theorem 4.1 (Implicit filtering property) *Suppose that in Theorem 3.1, Q and Z are computed using the BioGS algorithm on $F = T_k - \mu_Q I$ and $G = (T_k - \mu_Z I)^*$, such that (27) holds. If R_Q and R_Z have full rank, then*

$$\mathcal{R}(V_k^+) = \mathcal{R}((A-\mu_Q I)V_k) \tag{28}$$
$$\mathcal{R}(W_k^+) = \mathcal{R}((A-\mu_Z I)^* W_k). \tag{29}$$

Proof Let us prove the first statement (28). From (2) and (27), we derive

$$(A - \mu_Q I)V_k = V_k(T_k - \mu_Q I) + \beta_{k+1} v_{k+1} e_k^* = V_k Q R_Q + \beta_{k+1} v_{k+1} e_k^*. \quad (30)$$

Multiplying this on the right by R_Q^{-1} results in

$$(A - \mu_Q I) V_k R_Q^{-1} = V_k Q + \beta_{k+1}/\rho_Q v_{k+1} e_k^*, \quad (31)$$

where ρ_Q is the (k,k)-th element of R_Q. Since the tridiagonal matrix H is factorised as $H = R_Q Q$ (27), it holds that $h_{k,k-1} = \rho_Q q_{k,k-1}$, such that by (16)

$$(A - \mu_Q I) V_k R_Q^{-1} = \left[\begin{array}{cc} V_k Q I_{k,k-1} & V_k Q e_k + \beta_{k+1} q_{k,k-1}/h_{k,k-1} \end{array}\right] = \left[\begin{array}{cc} V_{k-1}^+ & \beta_k^+/h_{k,k-1} v_k^+ \end{array}\right], \quad (32)$$

from which (28) follows. The proof of (29) is analogous. □

Theorem 4.1 states that if the Implicitly Restarted Lanczos algorithm is repeated p times with shifts $\mu_{Q,1}, \ldots, \mu_{Q,p}$ and $\mu_{Z,1}, \ldots, \mu_{Z,p}$ using BioGS, then the subspace spanned by V_{k-p+1} is multiplied implicitly by a polynomial function $\phi(A) = (A - \mu_{Q,p} I) \cdots (A - \mu_{Q,1} I)$. Similarly, W_{k-p+1} is multipied by $\phi^{(*)}(A^*) = (A^* - \bar{\mu}_{Z,p} I) \cdots (A^* - \bar{\mu}_{Z,1} I)$. It should be noted that if this algorithm is applied on a symmetric matrix A and $v_1 = w_1$, such that $W_k = V_k$, then the filtering with $\mu_i \equiv \mu_{Q,i} = \mu_{Z,i}$ is still different from an IRA filtering. With IRA, the filtering function for V_k is $\phi(A)$, so the same holds for W_k – since both matrices are equal. However, unless the μ_i are real, the $\phi(A) \neq \phi^{(*)}(A^*) = \phi^{(*)}(A)$.

It can be shown for IRA [14, Lem. 3.10] that if the approximated spectrum is divided into two disjoint parts

$$\{\theta_1, \ldots, \theta_p\} \cup \{\theta_{p+1}, \ldots, \theta_k\} \quad (33)$$

and if the first set of p approximate eigenvalues is used as shifts $\mu_1 \equiv \theta_1, \ldots, \mu_p \equiv \theta_p$, then the approximate eigenvalues of the restarted relation will be given by $\{\theta_{p+1}, \ldots, \theta_k\}$. Also the approximate eigenvectors (left and right) will be the same. The first p approximate eigenvectors are said to be *deflated.*

Before we derive a similar property for the IRL algorithm, we propose a simple condition for deflation. Suppose that the subspaces (V_k, W_k) are restarted (or transformed) into (V_{k-1}^+, W_{k-1}^+). We say that the i-th approximate eigenpair (θ_i, y_i) is *not* deflated if *the vector that is removed from* V_k *is orthogonal to* y_i.

Lemma 4.2 *Say that the right approximate eigenpairs of* A *are given by* $(\theta_i, y_i = V_k u_i)$, $i = 1, \ldots, k$, *where* $\theta_i \neq 0$. *If* w, $v \in \mathbf{C}^n$ *are vectors such that* $\mathcal{R}\left([V_{k-1}^+, v]\right) = \mathcal{R}(V_k)$, $\mathcal{R}\left([W_{k-1}^+, w]\right) = \mathcal{R}(W_k)$, *and* $[W_{k-1}^+, w]^* [V_{k-1}^+, v] = I$, *then*

$$w \perp y_i \Leftrightarrow \text{ there exist a } u_i^+ \in \mathbf{C}^{k-1} \text{ such that } \quad T_{k-1}^+ u_i^+ - \theta_i u_i^+ = 0 \quad \text{and} \quad y_i = V_{k-1}^+ u_i^+, \quad (34)$$

i.e. w *is orthogonal to* y_i, *iff* (θ_i, y_i) *is an approximate eigenpair of the restarted relation. The same holds for the left approximate eigenpairs.*

Proof There exist transformation matrices P_V and P_W such that $V_k = [V_{k-1}^+, v] P_V$ and $W_k = [W_{k-1}^+, w] P_W$ – clearly $P_W^* P_V = I$. If we set $\tilde{u}_i = P_V^{-1} u_i = P_W^* u_i$, then $y_i = [V_{k-1}^+, v]\tilde{u}_i$ and

$$P_W^* T_k u_i = \theta_i P_W^* u_i \Rightarrow [W_{k-1}^+, w]^* A [V_{k-1}^+, v] \tilde{u}_i = \theta_i \tilde{u}_i. \quad (35)$$

By $[W_{k-1}^+, w]^* [V_{k-1}^+, v] = I$, we get $w^* [V_{k-1}^+, v] = e_k^*$. Thus, by multiplying with $\tilde{u}_i$,

$$w^* y_i = w^* [V_{k-1}^+, v] \tilde{u}_i = e_k^* \tilde{u}_i.$$

If $w^* y_i = 0$, then this means that we can rewrite

$$\tilde{u}_i = \left[\begin{array}{c} u_i^+ \\ 0 \end{array}\right], \quad (36)$$

and $y_i = V_{k-1}^+ u_i^+$. Because $T_{k-1}^+ = (W_{k-1}^+)^* A V_{k-1}^+$, it then follows by (35) that $T_{k-1}^+ u_i^+ - \theta_i u_i^+ = 0$. Inversely, if $\tilde{u}_i$ fulfils in (36), then it must follow that $w^* y_i = 0$, so $w \perp y_i$. □

We now show that if the Lanczos relations are restarted with BioGS and $\mu = \theta_j$, then the restarted subspace $\mathcal{R}\left(V_{k-1}^+\right)$ is spanned by the remaining approximate eigenvectors y_i, $i \neq j$, because it fulfils the conditions of Lemma 4.2.

	$\|\mathcal{P}^{\perp}_{V^+}(A-\mu I)V_k\|_2$	$\|\theta_2-\theta_2^+\|$	$\|AV_{k-1}^+ - V_{k-1}^+T_{k-1}^+ - \beta_k^+ v_k^+\|_2$
$\mu=\lambda_1$	$8e$–15	$3e$–3	$3e$–15
$\mu=\theta_1$	$3e$–15	$9e$–16	$3e$–15
$\mu=1.5$	$2e$–14	$3e$–1	$2e$–14

Table 1: Restarting the Lanczos algorithm using BioGS. For three different choices of μ (a true eigenvalue, an approximate eigenvalue and a 'random' number), the error on the implicit filter is shown ($\mathcal{P}^{\perp}_{V^+}(A-\mu I)V_k$ is the projection of $(A-\mu I)V_k$ on the nullspace of V_k^+). Also the difference between the old and the new (second) approximate eigenvalue is shown, as well as the error on the Lanczos equation.

Lemma 4.3 Exact shift property) *Say that the approximate eigenpairs of A are given by $(\theta_i, y_i = V_k u_i)$, $i = 1, \ldots, k$. If Theorem 3.1 is applied with BioGS and $\mu \equiv \theta_j$, $(1 \le j \le k)$, then the following statement is true : If $w \in \mathcal{R}(W_k)$ and $w \perp V_{k-1}^+$, then $w \perp y_i$, for i such that $\theta_i \ne \theta_j$.*

Proof If θ_j is an eigenvalue of T_k, then $(T_k - \theta_j I) = QR_Q$ does not have full rank and the (k,k)-th element of R_Q is zero (if another diagonal element were zero, the BioGS would have had a breakdown). Hence, $R_Q = I_{k,k-1}I_{k,k-1}^* R_Q$ and thus $V_k Q R_Q = V_k Q I_{k,k-1} I_{k,k-1}^* R_Q = V_{k-1}^+ I_{k,k-1}^* R_Q$.

Following Theorem 4.1, it holds that (see (31))

$$w^*(A-\theta_j I)V_k u_i = w^* V_k Q R_Q u_i + \beta_{k+1} w^* v_{k+1} e_k^* u_i = w^* V_{k-1}^+ I_{k,k-1}^* R_Q u_i + \beta_{k+1} w^* v_{k+1} e_k^* u_i = \beta_{k+1} w^* v_{k+1} e_k^* u_i. \tag{37}$$

On the other hand, it holds that (see (30))

$$w^*(A-\theta_i I)V_k u_i = w^* V_k (T_k - \theta_i I) u_i + \beta_{k+1} w^* v_{k+1} e_k^* u_i = \beta_{k+1} w^* v_{k+1} e_k^* u_i. \tag{38}$$

If we subtract both equations, then we find that $(\theta_j - \theta_i) w^* V_k u_i = (\theta_j - \theta_i) w^* y_i = 0$. Using Lemma 4.2, we get the result. □

If a pair of Lanczos relations is restarted with BioGS and with a shift equal to the approximate eigenvalue, then the approximate eigenvector is filtered out of the subspace basis. By analogy to IRA, we call this choice of shifts the use of *exact shifts*. If the Lanczos relations are restarted with p *exact shifts*, then the corresponding eigenpairs are removed from the relations.

Corollary 4.4 *Suppose that the spectrum of T_k is divided into two disjoint parts*

$$\{\theta_1, \ldots, \theta_p\} \cup \{\theta_{p+1}, \ldots, \theta_k\} \tag{39}$$

and suppose that the first set of p approximate eigenvalues is used as exact shifts for the implicitly restarting procedure. Then the right approximate eigenpairs of the resulting equations are given by $(\theta_i^+, y_i^+) = (\theta_{i+p}, y_{i+p})$, $i = 1, \ldots, k-p$. The same holds for the left approximate eigenpairs.

Proof The proof follows from Lemma 4.2 and 4.3, repeated p times. □

Example 4.1 Let us illustrate these results with a small example. Consider a Toeplitz matrix $A \in \mathbf{R}^{100\times 100}$ that is defined by

$$a_{i,i+k} = \frac{1}{k-1} \quad \text{for} \quad k \ge 0 \qquad \text{and} \qquad a_{i,i-k} = \frac{-1}{k-1} \quad \text{for} \quad k > 0.$$

All eigenvalues λ_i of A have real part equal to one : $\text{real}(\lambda_i) = 1$, $i = 1, \ldots, 100$. We performed 10 steps of Lanczos with $v_1 = e_1 = w_1$ and then restarted with BioGS and three different choices of $\mu = \mu_Q = \mu_Z$: a true eigenvalue $\mu = \lambda_1$, an approximate eigenvalue $\mu = \theta_1$ and a random number $\mu = 1.5$. The results are displayed in Table 1 The table illustrates Theorem 4.1 and Lemma 4.3 to computer precision. If BioGS is used, then the filtering property holds for all three cases. If an exact shift is used, then the corresponding approximate eigenvalue is removed from the spectrum of T^+, without changing the other approximate eigenvalues. We also used an inner Lanczos loop to restart the algorithm. The results for the error on the Lanczos relation were the same, but neither the implicit filtering property nor the exact shift property was fulfilled, as expected.

5 Implicitly Restarting and Breakdown

The Lanczos algorithm can break down. If this breakdown is not the result of the fact that some solution has been found, then it must be solved with look-ahead or it must be restarted completely. The main argument against an explicit restart is that it is expensive. It costs $2k$ matrix-vector multiplications and as many bi-orthogonalisation steps. Also, it can not guarantee that a breakdown will not reoccur in further iterations, although the same argument holds for look-ahead. In this section, we show how the implicit restart algorithm can be used to get around the serious breakdown problem.

Theorem 5.1 *Suppose that the Lanczos relations are restarted as in Theorem 3.1, and suppose that* $w_{k+1}^* v_{k+1} = 0$. *Then* $\beta_k^+ \gamma_k^+ (w_k^+)^* v_k^+ = h_{k,k-1} h_{k-1,k}$.

Proof Multiplying (16) by (17), results in

$$\begin{aligned}
\beta_k^+ \gamma_k^+ (w_k^+)^* v_k^+ &= (\bar{h}_{k-1,k} W_k Z e_k)^* (h_{k,k-1} V_k Q e_k) + (\bar{\gamma}_{k+1} z_{k,k-1} w_{k+1})^* (\beta_{k+1} q_{k,k-1} v_{k+1}) \\
&= h_{k-1,k} h_{k,k-1} e_k^* Z^* W_k^* V_k Q e_k + \gamma_{k+1} \beta_{k+1} \bar{z}_{k,k-1} q_{k,k-1} w_{k+1}^* v_{k+1} \\
&= h_{k-1,k} h_{k,k-1} + 0.
\end{aligned}$$

□

Theorem 5.1 shows that if H is unreduced, then no breakdown occurs in the next Lanczos step of the restarted method. If $h_{k,k-1} = 0$, then the last column of Q turns out to be an eigenvector of T_k. In this case, a new decomposition (12) must be found. Notice that this problem occurs with BioGS if the shift μ is chosen equal to an approximate eigenvalue θ_i. In that case, $T_k - \mu I$ is singular and the last vector of Q will be the null-eigenvector (it must be the last vector, because otherwise the BioGS algorithm breaks down).

The result of Theorem 5.1 is at least against our intuition. The restarting procedure applies a polynomial filter on the subspaces V_k and W_k, but the new Krylov subspaces will be subsets of the old subspaces, even when the algorithm proceeds. One could argue that, since breakdown seems to be a property of these Krylov subspaces, it can not be avoided with an implicit restart. In general, this is not true. The new search spaces are indeed very similar to the Krylov subspaces that would have been found if the algorithm proceeded, but each new pair of vectors is bi-orthogonalised to one vector less (i.e. $V_k Q e_k$ and $W_k Z e_k$). It is this 'missing' vector that prevents the new, bi-orthogonalised vectors to be orthogonal to each other. Indeed, the term $h_{k,k-1} h_{k-1,k}$ corresponds following (16) and (17) to the parts of v_k^+ and w_k^+ that contain traces of $V_k Q e_k$ and $W_k Z e_k$.

There is a relation between the moment matrix of restarted Lanczos relations and the moment matrix of the original relations. Suppose that in Theorem 4.1, $\mu \equiv \mu_Q = \mu_Z$, then the moment matrix of the restarted relation is $M_k(A, (A - \mu I)v_1, (A - \mu I)^* w_1)$. The rank of this matrix is equal to $\text{rank}(M_k(A - \mu I, (A - \mu I)v_1, (A - \mu I)^* w_1)) = \text{rank}(M_k^2(A, v_1, w_1))$. Indeed, the lower right $k \times k$ submatrix of $M_{k+1}(A - \mu I, v_1, w_1)$ is equal to the lower left submatrix of $M_{k+2}(A - \mu I, v_1, w_1)$, because of the Hankel structure, (see Figure 1). Therefore, implicitly restarting corresponds to shifting to a submatrix of the original moment matrix. In the following lemma, we show that if $M_{k+1}, \ldots, M_{k+p}$ are singular, then it takes at least p restarts to avoid the singularity. We only consider restarts with $\mu = \mu_Q = \mu_Z = 0$, because it is easier to understand. Since the rank of the moment matrix is equal for A and $A - \mu I$ (Lemma 3.5), it also holds for the general case.

Lemma 5.2 *Consider a matrix A and two starting vectors v_1, w_1. Suppose that all implicit restarts are computed with BioGS and with $\mu = 0$. If* $\text{rank}(M_k) = \text{rank}(M_{k+1}) = \cdots = \text{rank}(M_{k+p}) = k$, *then*

a) If l is such that $\text{rank}(M_{k+l}) = k + l$, *then $l \geq p$. If $i \leq p$, then* $\text{rank}(M_{k+p+i}) \leq \text{rank}(M_{k+p+i-1}) + 2$.

b) If for all $i = 1, \ldots, 2p-1$, it holds that $\text{rank}(M_{k+1}^i) < k + 1$ *and* $\text{rank}(M_{k+p+1}) \geq k + 1$, *then* $\text{rank}(M_{k+1}^{2p}) = k + 1$, *unless $e_i \in \mathcal{R}(M_{k+2p+1} I_{k+2p+1,2p})$, for some $1 \leq i \leq 2p$.*

Proof Consider the rectangular Hankel matrix $M_{k,l} \equiv M_k I_{k,l}$, $k \geq l$.
a) Since $M_{k+1}(A, v_1, w_1)$ differs from $M_k(A, v_1, w_1)$ in one row and one column, the rank of M_{k+1} can be at most $\text{rank}(M_k) + 2$. If the null space of M_{k+p} has dimension p, then it will take at least p new rows and p new columns in order to end up with a full rank moment matrix.
b) In [6, Cor. 5.1, p81], it is proven that

$$\begin{aligned}
\text{rank}(M_{k+2l,k}) &= \min\{k, \text{rank}(M_{k+l})\} &(40)\\
\text{rank}(M_{k+2l+1,k}) &= \min\{k, \text{rank}(M_{k+l+1,k+l})\} \leq \min\{k, \text{rank}(M_{k+l})\}. &(41)
\end{aligned}$$

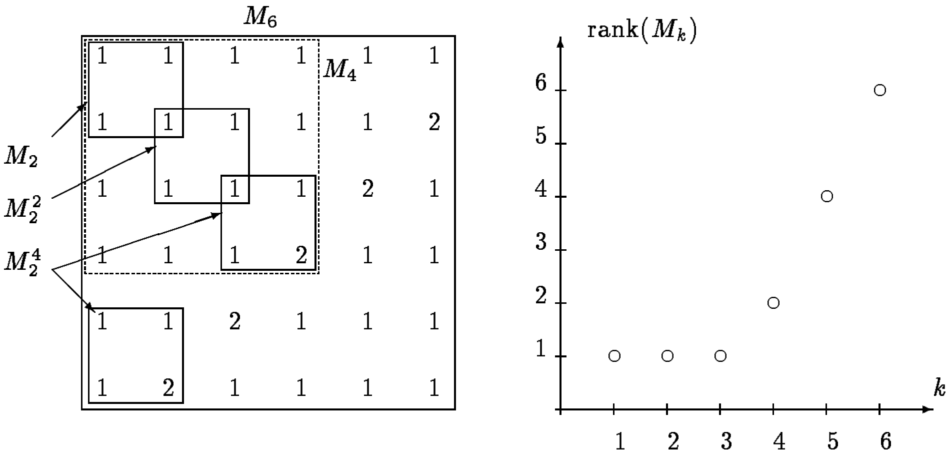

Figure 1: A connection between IRL and look-ahead Lanczos. The look-ahead Lanczos algorithm jumps from the (full rank) moment matrices M_1 to M_6. Implicitly restarted Lanczos results with each restart in a shifted moment matrix $M_2 \to M_2^2 \to M_2^4$, until the breakdown has disappeared. The circles in the right picture show the rank of the M_k matrices. The rank does not change from M_1 to M_3 and then grows with steps of one or two.

Since M_{k+1}^{2i} is the lower $k+1 \times k+1$ submatrix of $M_{k+2i+1,k+1}$, we can derive that

$$\mathrm{rank}(M_{k+1}^{2i}) \leq \mathrm{rank}(M_{k+2i+1,k+1}) = \min\{k+1, \mathrm{rank}(M_{k+i+1})\}.$$

For the same reason $\mathrm{rank}(M_{k+1}^{2i-1}) \leq \min\{k+1, \mathrm{rank}(M_{k+i+1})\}$. From (40), it follows that if $\mathrm{rank}(M_{k+i}) = k$, then $\mathrm{rank}(M_{k+2i-1,k+1}) = k$ and also $\mathrm{rank}(M_{k+2i,k+1}) = k$. On the other hand, $\mathrm{rank}(M_{k+2p+1,k+1}) = \min\{k+1, \mathrm{rank}(M_{k+p+1})\} = k+1$. If $e_i \notin \mathcal{R}(M_{k+2p+1,k+1})$ for $i = 1, \ldots, 2p$, then $\mathrm{rank}(M_{k+1}^{2p}) = \mathrm{rank}(M_{k+2p+1,k+1}) = k+1$, since

$$\left[\begin{array}{cc} M_{k+2p+1,k+1} & I_{k+2p+1,2p} \end{array} \right] = \left[\begin{array}{cc} \star & I_{2p} \\ M_{k+1}^{2p} & 0 \end{array} \right],$$

then has full rank. □

Lemma 5.2 proves two properties of a singularity in the Lanczos process. The first part shows that a singularity in a sequence of M_k matrices consists of a part where the rank is constant and a second part where the rank in general grows with 2 (unless there is an overlap of singularities or if the size of the singularity is odd). The second part proves that if a look-ahead algorithm has to compute a look-ahead step of length $2p$ or $2p-1$, then the corresponding implicitly restarted Lanczos method has to be restarted at least p times.

The connection of implicitly restarted Lanczos with look-ahead is illustrated in Figure 1. In the figure, $M_{k+1} = M_2$ is singular. It is easy to see that $M_2, \ldots, M_5$ are singular too, but $\mathrm{rank}(M_6) = 6$. A look-ahead algorithm has to take a step of length 5 in order to find a new pair of bi-orthogonal matrices. If IRL is used, then one iteration after the first restart, a new breakdown will be encountered, since $M_{k+1}^2 = M_2^2$ is singular. If the algorithm is then restarted again, then the moment matrix M_2^4 is no longer singular, and the algorithm can proceed without breakdown. So Look-ahead computes $M_2 \to M_3 \to M_4 \to M_5 \to M_6$, whereas IRL computes $M_2 \to M_2^2 \to M_2^4$. If a Lanczos algorithm encounters an incurable breakdown, so that there exist no look-ahead step that solves it, then IRL can not solve the breakdown either.

Example 5.1 We generate a set of matrices and starting vectors that lead to breakdown of the Lanczos algorithm as follows. Say that breakdown must occur at step m and that look-ahead must compute a step of length p. Consider then a random unreduced tridiagonal matrix B. If we perform Lanczos with $v_1 = e_1 = w_1$, then $V_k = I_{n,k} = W_k$. If we set $b_{m+1,m} = 0$ and $b_{m+p+1,m} = 1$, then it is easy to see that breakdown must

	$p=1$		$p=2$		$p=3$		$p=4$	
	l	$\|\|AV-VT\|\|$	l	$\|\|AV-VT\|\|$	l	$\|\|AV-VT\|\|$	l	$\|\|AV-VT\|\|$
IRL-BioGS	1	$2e$–11	1	$2e$–11	2	$1e$–10	2	$9e$–10
IRL-Lanczos	1	$5e$–11	1	$2e$–10	2	$2e$–10	2	$4e$–10
LA-Lanczos	1	$1e$–11	2	$2e$–12	3	$6e$–10	4	$1e$–10

Table 2: Comparison between IRL and look-ahead Lanczos for breakdown of different lengths ($p = 1, 2, 3, 4$) at step 5 of the algorithm. Shown is the number of required restarts (l, for LA-Lanczos this indicates the length of the look-ahead jump) and the overall error on the first Lanczos equation when the size of V_k and W_k is 10. The results are generated with Matlab4 on a DEC5000.

occur : $v_m = e_{m+p+1} \perp w_m = e_m$. The matrix can then be transformed, using a bi-orthogonal pair of matrices Q and $Z \in \mathbf{C}^{n\times n}$:

$$A = QBZ^*, \quad v_1 = Qe_1 \quad \text{and} \quad w_1 = Ze_1.$$

We generated four 100×100 matrices which suffered from breakdown at step $m = 5$, each with a growing singularity ($p = 1, 2, 3, 4$). Then we applied the following algorithms to these matrices : implicitly restarted Lanczos with BioGS ($\mu = \mu_Q = \mu_Z = 0$), IRL restarted with Lanczos ($\mu = 0$ and $q_1 = z_1 = (e_1 + e_2)/\sqrt{2}$) and look-ahead Lanczos. The results are shown in Table 2. The table illustrates the results of Lemma 5.2. The implicitly restarted Lanczos algorithm needs ceil($p/2$) restarts in order to handle the singularity. We also showed the error on the first Lanczos equation *after* the restart (i.e. after 10 iterations). The accuracy of all three applications is comparable, but it certainly depends on our implementation and on the parameters that are used for the restart. The optimal choice of the parameters, if it exists, should be the subject of further research.

6 Conclusions

This text generalises the concept of implicitly restarting an iterative algorithm to the Lanczos method for eigenvalue problems. We showed that a full run of the Lanczos algorithm on the small, projected eigenvalue problem generates a bi-orthogonal transformation for the Lanczos bases. The small Lanczos run can also be written as a BioGS algorithm, and connects the restarted subspaces with implicitly filtering of the original subspaces. The polynomials that define this filtering are governed by the starting vectors of the small Lanczos process or by the shifts μ_Q and μ_Z of the BioGS factorisation.

The resulting IRL algorithm can be used as an alternative for look-ahead in case of serious breakdown. IRL and look-ahead are connected in that if look-ahead extends the subspaces V_k and W_k with size p, then IRL selects in these large subspaces a set of k-dimensional subspaces V_k^+ and W_k^+ that do not suffer from breakdown. At the cost of ceil($p/2$) extra iteration steps, IRL returns a tridiagonalisation of A, whereas look-ahead Lanczos results in a *block* tridiagonal matrix. Both algorithms fail if incurable breakdown occurs.

The advantages of implicitly restarted Lanczos correspond to the advantages of implicitly restarted Arnoldi with respect to Arnoldi. If the Lanczos algorithm converges too slowly, then IRL can reduce the size of the subspaces V_k and W_k without losing too much information. If there are spurious eigenvalues (e.g. approximations of an infinite eigenvalue) or if a known, dominant eigenvalue must be avoided (e.g. a zero eigenvalue), then IRL can be used to *filter* away these eigenvalues. Finally, it is an alternative for look-ahead in case of breakdown. It has the additional advantage that even in case of near-breakdown, IRL can be employed and the tridiagonal structure of the T_k matrix is preserved. So the decision 'is this a breakdown or not ?' is less important when IRL is used, since the result is tridiagonal anyway – look-ahead makes the matrix block tridiagonal. The disadvantage of implicitly restarted Lanczos is that it costs one iteration step per restart, since IRL reduces the size of the approximating subspaces with one. It is also possible that the restarting algorithm itself incorporates some of the disadvantages of the Lanczos algorithm, such as possible instabilities or inaccurate results due to a near-breakdown.

Acknowledgments. This research was supported by the National Fund for Scientific Research (FWO), project "Orthogonal systems and their applications", grant #G.0278.97.

This paper presents research results of the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture. The scientific responsibility rests with its authors.

References

[1] C. Brezinski, M. Redivo Zaglia, and H. Sadok. A breakdown free Lanczos type algorithm for solving linear systems. *Numer. Math.*, 63(1):29–38, 1993.

[2] D. Calvetti, L. Reichel, and D.C. Sorensen. An implicitly restarted Lanczos method for large symmetric eigenvalue problems. *ETNA*, 2:1–21, 1994.

[3] R.W. Freund, M.H. Gutknecht, and N.M. Nachtigal. An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices, Part I. *SIAM J. Sci. Statist. Comput.*, 14(1):137–158, 1993.

[4] E.J. Grimme, D.C. Sorensen, and P. Van Dooren. Model reduction of state space systems via an implicitly restarted Lanczos method. *Num. Alg.*, 12:1–32, 1996.

[5] M.H. Gutknecht. Lanczos-type solvers for nonsymmetric linear systems of equations. *Acta Numerica*, 6:271–397, 1997.

[6] G. Heinig and K. Rost. *Algebraic Methods for Toeplitz-like Matrices and Operators*, volume 19 of *Mathematical Research*. Akademie-Verlag Berlin, 1984. 212p.

[7] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Nat. Bur. Stand.*, 45:255–281, 1950.

[8] R.B. Lehoucq, D.C. Sorensen, and P. Vu. ARPACK: *An implementation of the Implicitly Re-started Arnoldi Iteration that computes some of the eigenvalues and eigenvectors of a large sparse matrix.* Available from netlib@ornl.gov under the directory scalapack, 1995.

[9] R.B. Morgan. On restarting the Arnoldi method for large nonsymmetric eigenvalue problems. *Math. Comp.*, 65:1213–1230, 1996.

[10] C. Paige, B.N. Parlett, and H.A. Van der Vorst. Approximate solutions and eigenvalue bounds from Krylov subspaces. *Num. Lin. Alg. Appl.*, 2:115–133, 1995.

[11] B.N. Parlett. Reduction to tridiagonal form and minimal realizations. *SIAM J. Matrix Anal. Applic.*, 13(2):567–593, apr 1992.

[12] B.N. Parlett, D.R. Taylor, and Z.A. Liu. A Look-Ahead Lanczos Algorithm for Unsymmetric Matrices. *Math. Comp.*, 44:105–124, 1985.

[13] Y. Saad. Variations on Arnoldi's Method for Computing Eigenelements of Large Unsymmetric Matrices. *Linear Alg. Appl.*, 34:269–295, 1980.

[14] D.C. Sorensen. Implicit application of polynomial filters in a k-step Arnoldi method. *SIAM J. Matrix Anal. Applic.*, 13:357–385, 1992.

[15] A. Stathopoulos, Y. Saad, and K. Wu. Dynamic thick restarting of the Davidson, and the Implicitly Restarted Arnoldi methods. *SIAM J. Sci. Comput.*, 1997. To appear.

[16] D.R. Taylor. *Analysis of the Look Ahead Lanczos Algorithm.* Ph.d. thesis, University of California, Berkeley, 1982.