Implicitly Restarting Lanczos

Gorik De Samblanx Adhemar Bultheel

Abstract

We show in this text how the idea of the Implicitly Restarted Arnoldi method can be generalised to the non-symmetric Lanczos algorithm, using the two-sided Gram-Schmidt process or using a 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.

KEYWORDS: non-symmetric 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.

For Arnoldi's method, the idea of *explicitly* restarting the factorisation was introduced by Saad [10] and different strategies were formulated on how the restart must be applied in practice. A break-through was accomplished with the *implicitly* restarted Arnoldi method (IRA) of Sorensen [11], which lead to the popular ARPACK eigenvalue package [7].

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. 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.

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. [5] 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.

put here title
put here (eds.), pp. 1–5.
Copyright ©put here year by IMACS
All rights of reproduction in any form reserved.
put here ISBN

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 and the column range of a matrix is denoted by $\mathcal{R}(Q)$.

2 BioGS versus Lanczos

Before we show how to restart the Lanczos algorithm, we describe the algorithm itself [6]. There are many different implementations possible, which are all equivalent upto some scaling of the Lanczos vectors. The restarting procedure (Section 3) does not depend on the exact implementation of the algorithm, so we may are free to choose one.

We also recall the two sided Gram-Schmidt algorithm as presented by Parlett [8]. Both algorithms can be seen as algorithms that compute a set of bi-orthogonal bases for the ranges of a given pair of matrices.

Algorithm 1 *Simple Lanczos*
In: $A \in \mathbf{C}^{n\times n}$, $v_1, w_1 \in \mathbf{C}^n$, k
Out: $V_k, W_k \in \mathbf{C}^{n\times k}$, $\alpha_i, \beta_i, \gamma_i$
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*
   $v_{j+1} = v_+/\beta_{j+1}$ *and* $w_{j+1} = w_+/\bar{\gamma}_{j+1}$
   with $\beta_{j+1}\gamma_{j+1} = w_+^* v_+$.
   Else Breakdown and Stop.

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* $j = 2, \ldots, k$ *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*
   $q_j = f_j/\rho_Q$ *and* $z_j = g_j/\rho_Z$
   with $\bar{\rho}_Z \rho_Q = g_j^* f_j$
   Else Breakdown and Stop

The non-symmetric Lanczos algorithm attempts to compute a tridiagonal factorisation of a square matrix A. Unless it breaks down, the algorithm computes a bi-orthogonal pair of basis matrices (V_k, W_k) and a full rank tridiagonal matrix T_k such that

$$AV_k = V_k T_k + \beta_{k+1} v_{k+1} e_k^* \tag{1}$$

$$A^* W_k = W_k T_k^* + \bar{\gamma}_{k+1} w_{k+1} e_k^*, \tag{2}$$

$$\text{with} \quad W_k^* V_k = I \qquad \text{and } T_k = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \gamma_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \gamma_{k-1} & \alpha_{k-1} & \beta_k \\ & & & \gamma_k & \alpha_k \end{bmatrix}. \tag{3}$$

Equations (1) and (2) are called the *Lanczos relations*. They can be seen as approximate Schur decompositions of A and A^*, from which approximate eigenvalues and eigenvectors are computed as follows. A pair (θ_i, y_i) is an approximate (right) eigenpair of A if $y_i = V_k u_i$ where u_i solves $T_k u_i - \theta_i u_i = 0$. The corresponding (right) residual can be computed as $\|Ay_i - \theta_i y_i\|_2 = |u_{i,k}\beta_{k+1}|\|v_{k+1}\|_2$. The algorithm has converged if $|u_{i,k}\beta_{k+1}|\|v_{k+1}\|_2 \simeq 0$.

It is well known that the Lanczos algorithm might suffer from breakdown. Breakdown occurs whenever $w_+^* v_+ = 0$ in the algorithm, 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}. If $v_+ = 0$ or $w_+ = 0$, then it we have a *lucky* breakdown, since this means that an invariant subspace for A is found. Otherwise, the breakdown is called *serious*. If $w_+^* v_+ \simeq 0$, then a *near* breakdown occurs (which may cause numerical instabilities).

Notice that if A is tridiagonal and the starting vectors can be written as $v_1 = v_{1,1}e_1 + v_{1,2}e_2$ and $w_1 = w_{1,1}e_1 + w_{1,2}e_2$, then both V_k and W_k will be upper Hessenberg. Moreover, if $v_1 = v_{1,1}e_1$, then V_k will be upper triangular and W_k lower bi-diagonal.

Algorithm 2 shows an implementation of the bi-orthogonal or two sided Gram-Schmidt process (BioGS). We assume that this algorithm will not break down. Given a pair of matrices (F, G), BioGS returns a pair of bi-orthogonal matrices (Q, Z) and a pair of upper triangular matrices (R_Q, R_Z) which fulfill a set of QR-type relations

$$F = Q R_Q \text{ and } G = Z R_Z. \tag{4}$$

If F and G have full rank, than R_Q and R_Z will be invertible.

Theorem 2.1 *Suppose that $F = G^* \in \mathbf{C}^{k\times k}$ are full rank tridiagonal matrices, which are decomposed in (4) with the BioGS algorithm. If (V_k, W_k) are the results of the Lanczos algorithm applied to F with $v_1 = Fe_1$ and $w_1 = F^* e_1$, then there exists a diagonal scaling matrix D such that $Q = V_k D$ and $Z = W_k D^{-1}$.*

Proof : See [3]. $\diamond$

Theorem 2.1 shows that for a tridiagonal matrix, the BioGS algorithm can be treated as a special case of the Lanczos algorithm. In practice however, it should be implemented as in Algorithm 2. This algorithm seems to be numerically more stable, since it performs in each step a full orthogonalisation of the vectors f_j and g_j.

3 Implicitly Restarted Lanczos

Let us now show how the Lanczos algorithm can be restarted. We demonstrate this first for only one Lanczos relation, without using a shift. Given the relation (1) and suppose that we have a pair of square bi-orthogonal Hessenberg matrices (Q, Z) that fulfill

$$Q^* Z = I = Q Z^*, \;\; T_k Q = Q H \text{ and } T_k^* Z = Z H^*, \tag{5}$$

for some tridiagonal matrix $H \in \mathbf{C}^{k\times k}$. These matrices can be the result of a full Lanczos process. By multiplying (1) on the right by Q, we get

$$A V_k Q = V_k T_k Q + \beta_{k+1} v_{k+1} e_k^* Q = V_k T_k Q + \beta_{k+1} v_{k+1} (q_{k,k-1} e_{k-1}^* + q_{k,k} e_k^*). \tag{6}$$

If we insert $I = QZ^*$, then we find (Q is Hessenberg)

$$A(V_k Q) = (V_k Q)(Z^* T_k Q) + [0 \cdots 0, \; \beta_{k+1} q_{k,k-1} v_{k+1}, \; \beta_{k+1} q_{k,k} v_{k+1}]. \tag{7}$$

If we now remove the last column of this equation and define $V_{k-1}^+ = V_k Q I_{k,k-1}$ and $T_{k-1}^+ = I_{k,k-1}^* H I_{k,k-1}$ then we can derive from relation (7) that

$$A V_{k-1}^+ = V_{k-1}^+ T_{k-1}^+ + \beta_k^+ v_k^+ e_{k-1}^* \tag{8}$$

with (reorganising the equation)

$$\beta_k^+ v_k^+ = \beta_{k+1} q_{k,k-1} v_{k+1} + h_{k,k-1} V_k Q e_k. \tag{9}$$

Analogously, we find by defining $W_{k-1}^+ = W_k Z I_{k,k-1}$, the new Lanczos relation

$$A^* W_{k-1}^+ = W_{k-1}^+ (T_{k-1}^+)^* + \bar{\gamma}_k^+ w_k^+ e_{k-1}^*, \tag{10}$$

where

$$\bar{\gamma}_k^+ w_k^+ = \bar{\gamma}_{k+1} z_{k,k-1} w_{k+1} + \bar{h}_{k-1,k} W_k Z e_k. \tag{11}$$

Since $Z^* Q = I$, the new matrices V_{k-1}^+ and W_{k-1}^+ are bi-orthogonal :

$$(W_{k-1}^+)^* V_{k-1}^+ = I_{k,k-1}^* Z^* W_k^* V_k Q I_{k,k-1} = I_{k-1}, \tag{12}$$

so if T_{k-1}^+ is a full rank tridiagonal matrix, then these relations are correct Lanczos relations.

The Lanczos relations (1) and (2) can be shifted with a scalar μ before restarting it. This *shift* μ can then be used as an extra degree of freedom for the algorithm.

Theorem 3.1 *Given the Lanczos relations (1) and (2) and suppose that the upper Hessenberg matrices (Q, Z) and the tridiagonal matrix H are computed by a Lanczos algorithm, such that*

$$(T_k - \mu I)Q = QH \text{ and } (T_k - \mu I)^* Z = ZH^* \tag{13}$$

then the matrices $V_{k-1}^+ = V_k Q I_{k,k-1}$, $W_{k-1}^+ = W_k Z I_{k,k-1}$ and $T_{k-1}^+ = I_{k,k-1}^* (H + \mu I) I_{k,k-1}$ define a restarted Lanczos relation. Moreover, if (13) is computed by the BioGS algorithm, then

$$\mathcal{R}\left(V_{k-1}^+\right) = \mathcal{R}\left((A - \mu I)V_{k-1}\right) \text{ and } \mathcal{R}\left(W_{k-1}^+\right) = \mathcal{R}\left((A - \mu I)^* W_{k-1}\right). \tag{14}$$

If $\mu = \theta_j$, then $\mathcal{R}\left(V_{k-1}^+\right) = \mathcal{R}\left\{y_i \mid i \neq j\right\}$.

Proof: See [3] $\diamond$

The shift μ can thus be used to control the filter that is applied implicitly in (14), or to remove an approximate eigenvector from the subspace bases.

3.1 Restart in case of breakdown

When the Lanczos algorithm suffers from serious breakdown, then it must be continued in a different way. One can restart the method explicitly, which is expensive and does not guarantee to avoid a new breakdown, or one can give up the tridiagonal form of T_k and perform a Look-ahead strategy, e.g. [1, 4, 9]. An implicitly restarted Lanczos algorithm offers an alternative for look-ahead. The next theorem shows that by restarting the Lanczos relations, the breakdown should disappear.

Theorem 3.2 *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 : The proof can easily be seen from (9) and (11), combined with (12). $\diamond$

Theorem 3.2 shows that if H is unreduced, then no breakdown occurs in the next Lanczos step of the restarted method. If $h_{k-1,k} = 0$, then the last column of Q is an eigenvector of T_k. In that case, a new decomposition (5) must be found.

4 Conclusions

We proposed an algorithm that restarts the non-symmetric Lanczos algorithm implicitly. If BioGS is used, then the algorithm can be seen as a bi-orthogonal extension of the 'Sorensen-restart' for the symmetric Lanczos algorithm. Using the Lanczos algorithm itself for the restart, defines a restart that is even more general. The restarting scheme can be used for almost the same reasons as the Implicitly Restarted Arnoldi method. Additionally, it can cure a breakdown in the main Lanczos loop. Therefore, an Implicitly Restarted Lanczos method can result in a true tridiagonalisation of the original matrix, where a look-ahead Lanczos implementation returns a block-tridiagonal result. However, the definitive avoidance of breakdown can not be guaranteed. The restart also comes with an additional amount of computational work. Finally, care must be taken of the inner Lanczos or BioGS loop. The ordinary Lanczos algorithm can be numerically unstable, and the process in the inner loop might inherit this instability.

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

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] G. De Samblanx and B. Adhemar. Nested Lanczos : Implicitly Restarting a Lanczos algorithm. TW-report, Dept. Comp. Science, K.U.Leuven, aug 1997.

[4] 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.

[5] 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.

[6] 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.

[7] 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, directory scalapack, 1995.

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

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

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

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

Gorik De Samblanx, Adhemar Bultheel
K.U. Leuven, Dept. of Computer Science, Celestijnenlaan 200A, B-3001 Heverlee, Belgium
{Gorik.DeSamblanx Adhemar.Bultheel}@cs.kuleuven.ac.be