Implementation Aspects of Band Lanczos Algorithms for Computation of Eigenvalues of Large Sparse Symmetric Matrices

By Axel Ruhe*

Abstract. A band Lanczos algorithm for the iterative computation of eigenvalues and eigenvectors of a large sparse symmetric matrix is described and tested on numerical examples. It starts with a p dimensional subspace, and computes an orthonormal basis for the Krylov spaces of A, generated from this starting subspace, in which A is represented by a $2p + 1$ band matrix, whose eigenvalues can be computed. Special emphasis is given to devising an implementation that gives a satisfactory numerical orthogonality, with a simple program and few arithmetic operations.

1. Introduction. The Lanczos algorithm [7] for the computation of eigenvalues and eigenvectors of a real symmetric matrix has been subject to a great deal of analysis and development in the last few years. Paige [10], [11] gave a rounding error analysis, and advocated its use as an iterative algorithm, and its virtues and limitations were elucidated by Kahan and Parlett [6]. After the improvements proposed by Parlett and Scott [12], the algorithm is now on the verge of gaining acceptance as a standard piece of mathematical software.

The standard Lanczos algorithm, applied to the problem

$$(1.1) \qquad Ax = \lambda x,$$

starts with a single vector s, and works entirely in the Krylov subspace for A generated by that vector,

$$(1.2) \qquad K_j(s) := \text{span}\{s, As, \ldots, A^{j-1}s\}.$$

A natural extension is now to start with several starting vectors in parallel, yielding a subspace Lanczos algorithm. This idea has been attributed to several sources, and block Lanczos algorithms have been described and used by Cullum and Donath [2], [3], Golub and Underwood [5], [16] and Lewis [8]. They are also given consideration in the general treatment in [6].

The main reason for introducing a subspace Lanczos algorithm is the need to get several independent vectors corresponding to multiple eigenvalues, and speed the convergence when clustering occurs. In the context of inverse eigenvalue problems, one

---

needs a subspace Lanczos algorithm to produce a band matrix having prescribed eigen-values, see [1].

It is the purpose of the present contribution to show that a Lanczos algorithm that starts with several vectors is much more simply described as a band Lanczos algorithm. This description has the advantage that it is possible to use the analysis for the modified Gram Schmidt algorithm as given in a recent contribution by Daniel, Gragg, Kaufman and Stewart [4], to investigate the need for reorthogonalization at different steps. Further, the program is shorter and simpler; and the case when the dimension of the subspace is reduced, due to linearly dependent vectors, is simple to treat by band width reduction.

It might be noted that we started to work on band Lanczos algorithms earlier (see [13], [14], [15]), but that only the tests for orthogonality using [4] developed this into a workable program.

We start in Section 2 by describing the basic algorithm, and discussing how orthogonalization is to be done, and how stopping criteria can be implemented. We continue with a brief discussion of the choice of band width in Section 3, and con-clude with describing a numerical example in Section 4.

We will use the following notation for submatrices. If A is an $n \times n$ matrix, we let A_j be the $n \times j$ matrix of the first columns of A and A_{jj} be the leading $j \times j$ sub-matrix of A. The rest of the notation is standard.

2. Formulation of the Algorithm. The band Lanczos algorithm can be formulated concisely in the following way:

1. *Start with p orthonormal vectors forming*

(2.1)
$$Q_p = (q_1, \ldots, q_p).$$

2. *For each $j = 1, 2, \ldots$ until* convergence
   1. *Form $y := Aq_j$.*
   2. *Orthogonalize against $q_1, \ldots, q_{j+p-1}$, i.e. write*

(2.2)
$$Aq_j = Q_{j+p-1}t_j' + y',$$

   with $y'^T Q_{j+p-1} = 0$.
   3. *If $\|y'\|_2 >$ tolerance, then*

(2.3)
$$\text{Form } q_{j+p} := y'/(t_{j+p, j} := \|y'\|_2),$$

   else
   $$\text{Let } p := p - 1. \text{ If } p = 0 \text{ then convergence.}$$

Let us see what happens. Combining (2.2) and (2.3), we get $Aq_j = Q_{j+p}t_j$, which we combine with the corresponding expressions for $i = 1, 2, \ldots, j-1$ to get $AQ_j = Q_{j+p}T_j$.

The upper $j \times j$ matrix T_{jj} has at most p diagonals below the main diagonal filled, the number p possibly decreasing as we go along down the main diagonal.

However, since T_{jj} is the section of the symmetric matrix A, i.e., $T_{jj} = Q_j A Q_j^T$, it is symmetric and, thus, a band matrix of width $2p + 1$.

Orthogonalization. If we succeed in giving Q_j orthonormal columns, we only need to compute the elements on and below the main diagonal of T as orthogonaliza-

tion coefficients; those above may be obtained by symmetry. In a numerical computation we may, however, be interested in getting a better orthogonality of the basis vectors by reorthogonalizing against older vectors, or all the way back to the first column. We get the following picture of T.

(2.4) $$T =$$
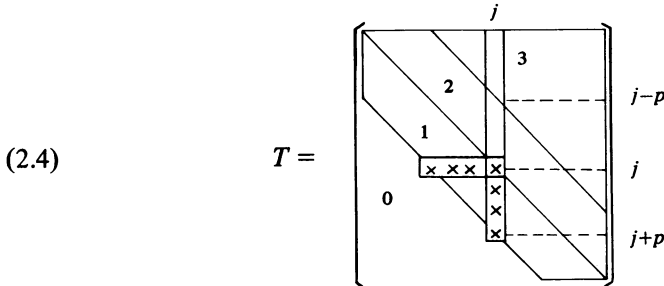
0 zeros by construction, not yet computed vectors

1 new orthogonalization coefficients, newer vectors

2 obtained by symmetry, older vectors

3 should be zero, oldest vectors

The most natural algorithm to use when performing the orthogonalization step 2.2 is modified Gram Schmidt (MGSA). Our computation is a special case of updating; at each step we add a new vector to a set of orthonormal vectors. If complete reorthogonalization is not performed, we also drop the oldest vector from the set, and continue to march along.

The rounding error analysis in [4] shows that MGSA will give satisfactory orthogonality, provided that a second orthogonalization step is performed, in case a substantial reduction in size occurred during orthogonalization. This means that, with v the vector to be made orthogonal and v' the resulting orthogonal vector, a reorthogonalization step is invoked whenever

(2.5) $$\theta \|v'\|_2 < \|v\|_2 ,$$

where θ is a tolerance. A value around $\theta = \sqrt{2}$ is advocated for full orthogonality, we have tested both $\theta = 1.4$ and the less stringent $\theta = 10$.

Besides the choice of reorthogonalization tolerance, we have also tested whether the elements above the diagonal should be obtained by symmetry or if it is of advantage to get them as orthogonalization coefficients, thus getting a better orthogonality of Q at the price of a slightly nonsymmetric matrix T. We did not test full reorthogonalization (1, 2, 3 in (2.4)), since that algorithm needs an order of magnitude more work and is already known to give good orthogonality in all cases (see Paige [9] or Underwood [16]).

We have tested an algorithm producing a nonsymmetric T by orthogonalizing against $2p$ vectors (2, 3 in (2.4)). With the reorthogonalization tolerance $\theta = 10$, we never needed an extra orthogonalization step, and the basis vectors lost orthogonality completely; not even the running basis of the last p vectors was orthogonal after a while. A drawback of the MGSA is that it does not discover that the set of supposedly

orthogonal vectors, against which the new vector is orthogonalized, has lost orthogonality, causing the new vector to be even more nonorthogonal. An interesting observation during these tests was also that it was only the outermost diagonal that was nonsymmetric, the inner band was perfectly symmetric in all cases. If we choose the smaller reorthogonalization tolerance $\theta = 1.4$, we needed a reorthogonalization step in every iteration, effectively doubling the operation count. Now we got both the symmetry and the orthogonality perfect. It was observed that both symmetry and orthogonality broke down at the same time, this fact is reason enough to discard the nonsymmetric variant of the algorithm.

When we tested the symmetric algorithm, making use of already produced T elements when performing the first orthogonalization to older vectors (2 in (2.4)), a similar pattern emerged. The choice $\theta = 10$ gave almost no reorthogonalizations, and complete loss of orthogonality even between neighboring vectors, while $\theta = 1.4$ needed reorthogonalization in every step. During most of the tests we noticed, however, that the band matrix T produced always had its larger elements in the main diagonal and in the outermost diagonal, while those diagonals in between contained considerably smaller elements. It seems that, although any band matrix may occur, those with three dominating diagonals are more likely to occur than others. The elements in the main diagonal are obtained by subtracting the Rayleigh quotient multiple from the operated vector. We modified the algorithm so that this was done explicitly at first, thus always giving an extra reorthogonalization against that vector. The outer diagonal is obtained by normalizing the vector and does not take part in the reorthogonalization process. With this modification we needed only few reorthogonalizations even if we set $\theta = 1.4$, so this is the recommended algorithm, giving both safety and economy. From our tests it appeared that reorthogonalization against older vectors (2 in (2.4)) did not gain much, in spite of a relatively high cost, the important fact is that the orthogonality to the newer vectors (1 in (2.4)) is perfect.

Convergence Criteria. The orthogonality to the old vectors (3 in (2.4)) is governed by a difference equation which is essentially similar to the one derived in [6]. Loss of orthogonality occurs when eigenvectors converge, and a band version of the truncation criterion given in [6] is used intermittently to check for convergence. Let z be an eigenvector of T_{jj}. The corresponding eigenvector (Ritz approximation) for A is $y = Q_j z$, and its residual is (assuming orthogonality of Q_j)

$$r = Ay - \lambda^{(j)} y$$

$$= AQ_j z - Q_j \lambda^{(j)} z = AQ_j z - Q_j T_{jj} z$$

$$= \left[AQ_j - Q_{j+p} T_j \right] z + Q_{j+p} \left[\begin{matrix} 0 \\ T_{pj} \end{matrix} \right] z,$$

giving (since the first term is zero)

$$\|r\|_2 = \|T_{pj} z\|_2 = \left(\sum_{i=j+1}^{j+p} \left(\sum_{k=i-p}^{j} t_{ik} z_k \right)^2 \right)^{1/2},$$

which is easy to compute since all elements of the lower $p \times j$ submatrix are available after step j. Note that only few elements of this submatrix are filled.

When one or a few eigenvector approximations have converged, we have to keep the later basis vectors orthogonal to them by periodic reorthogonalization as described by Parlett and Scott [12]. This amounts to adding blocks of elements in the upper (3 in (2.4)) part of T corresponding to the converged vectors regarded as linear combinations of basis vectors:
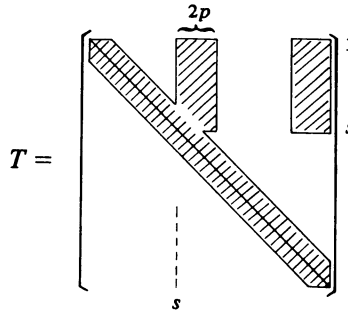
Here s is a point where an eigenvector has converged.

3. Choice of Band Width. Two needs govern the choice of band width p: the need for reliability, i.e., we want to avoid missing an eigenvalue of interest to us, and the need for economy, i.e., we want to have convergence with as little arithmetical work as possible. We have to remember that an algorithm using a larger band width is slower, both when forming T and when computing the eigenvalues of the band matrix.

The convergence of the eigenvalues of T_{jj} towards those of A is majorized by a set of orthogonal polynomials, and the fundamental result for the band case has been given by Underwood [16, Theorem 2.6.1].

With bandwidth $2p + 1$, perform $j = s \times p$ steps. Then it is true that

$$(3.1) \qquad \lambda_k(A) \leqslant \lambda_k(T_{jj}) \leqslant \lambda_k(A) + \epsilon_k^2, \quad \epsilon_k^2 = (\lambda_n - \lambda_k) \frac{\tan^2 \theta}{T_{s-1}^2((1 + \gamma_k)/(1 - \gamma_k))},$$

where

> θ *is the maximum angle between* span$\{Q_p\}$ *and p eigenvectors,*
>
> $(3.2) \qquad \gamma_k = (\lambda_k - \lambda_{p+1})/(\lambda_k - \lambda_n)$ *is the relative separation, and*
>
> T_{s-1} *is the Chebyshev polynomial.*

It should be emphasized that this is truly a worst case result, and that one in practice gets much better correspondence between the eigenvalues of T_{jj} and those of A. It is, e.g., possible to replace span$\{Q_p\}$ in (3.2) by any subspace of it, yielding a smaller angle θ, at the expense of a smaller separation γ_k. We always get at least as good an approximation as we would have got, had we started the ordinary tridiagonal Lanczos algorithm with the best vector available from span$\{Q_p\}$. Moreover, only

the case when $j \ll n$ is covered, when j approaches n, we are likely to exhaust R^n and get even better approximations.

We see from the a priori bound (3.1) and practical experience that the only case when the band width is of importance is when the interesting extreme eigenvalues are coincident or severely clustered. We are likely to get exactly p independent vectors corresponding to such a cluster, further vectors occurring only when convergence has occurred to a greater accuracy than the separation between λ_p and λ_{p+1}. Therefore, p should be chosen as the number of independent eigenvectors sought to any cluster of interesting eigenvalues.

A competing strategy to find several independent eigenvectors is the implicit deflation method proposed by Parlett and Scott [12].

4. A Numerical Example. Test runs with a program implementing the algorithm described in this paper have been performed on a Burroughs B6700 computer at the University of California, San Diego Computing Center. For the orthogonalization in step 2.2 of the algorithm, we used the Modified Gram-Schmidt algorithm given in [4]; and for the computation of eigenvalues and eigenvectors of T_{jj} we used the procedures *bqr* and *symray* from [17].

We report here results on a test matrix that we expect behaves as typical difference approximations to elliptic problems do. The test matrix was a diagonal matrix with the eigenvalues

$$\lambda_{jk} = \sin^2 \frac{j\pi}{2(m+1)} + \sin^2 \frac{k\pi}{2(m+1)}, \quad j, k = 1, 2, \ldots, m,$$

of the 5-point difference approximation to the Laplace equation over a square. It is evident that it has several double eigenvalues, so a choice of $p = 2$ is natural. We have tried $p = 1, 2, 3$ and 4.

The results are summarized in Table 4.1. Much of the discussion in Section 2 on orthogonalization strategies is based on these results. We have tried the two reorthogonalization tolerances $\theta = 1.4$ and $\theta = 10$ for determining when a reorthogonalization should take place (2.5). Further, we have used the six different strategies specified in Table 4.2.

It is noteworthy that the stringent choice of tolerance ($\theta = 1.4$) demands reorthogonalization in every step, but that the extra orthogonalization against q_j performed in strategies 4–6 obviates the need for almost all of these reorthogonalizations. The unsymmetric variants 3 and 6 gave no significantly better results than the symmetric 1, 2, 4, 5, and those that reorthogonalized against older vectors (2, 3, 5, 6) gave a slightly better result than those that did not (1, 4). If we take the operation counts into account, we, therefore, recommend strategy 4, which will give a satisfactory result with the least amount of computational work.

TABLE 4.1. Summary of results of test runs on band Lanczos algorithm.
Band width $2p + 1, p = 2$ when nothing else stated.
10 smallest eigenvalues sought. Order $n = 100$.

Strategy		Convergence of eigenvalues			Orthogonality of basis vectors			# vectors q_j such that $|q_{80}^T q_j| < 10^{-4}$
		steps j needed for $\lambda_1\ \lambda_2\ \lambda_3\ \lambda_4$	λ_{10}	Largest residual	# reort	$\max|q_{40}^T q_j|$	$\max|q_{60}^T q_j|$	
No	1	53 53 63 63	$83^{1)}$	4.0 (-7)	83	1.2 (-8)	1.6 (-2)	34
periodic	2	60 70 70 70	80	4.0 (-7)	80	4.5 (-9)	4.6 (-5)	31
reort	3	59 59 59 69	1.)	1.3 (-7)	79	6.0 (-9)	4.2 (-5)	33
$\theta = 1.4$								
With	1	60 60 60 70	80	4.1 (-7)	80	3.9 (-9)	3.1 (-6)	34
periodic	2	60 60 60 70	80	9.2 (-8)	80	5.5 (-9)	3.3 (-5)	31
reort	3	60 60 70 70	80	4.8 (-8)	80	3.4 (-9)	1.1 (-5)	30
	4	60 60 60 70	80	5.2 (-8)	4	6.8 (-9)	9.7 (-5)	37
$\theta = 1.4$	5	60 60 60 70	80	3.3 (-7)	0	8.1 (-9)	3.1 (-5)	35
	6	59 59 59 69	$79^{2)}$	$1.2(-8)^{2)}$	54	8.5 (-8)	1.6 (-4)	30
No	1	no convergence$^{3)}$	–	–	0	5.0 (-1)	5.4 (-1)	–
reort	2	no convergence$^{4)}$	–	–	0	9.2 (-1)	——	–
$\theta = 10$	3	60 60 60 –	–	–	0	3.8 (-3)	8.9 (-1)	–
Other	$p = 1$	30 42 5) 42	5)	5)	0	2.4 (-1)	2.9 (-1)	–
band	$p = 3$	70 70 70 75	90	1.6 (-7)	6	3.1 (-8)	1.6 (-5)	41
widths. –	$p = 4$	75 75 80 80	96	1.6 (-6)	11	1.3 (-9)	3.4 (-7)	31
$\theta = 1.4$ strat = 4								

1) Copies of already converged eigenvalues occurred   3) Elements bounded but orthogonality lost
2) Only 9 were converged here   4) Elements of T unbounded in size
                                                         5) Only one eigenvalue from each pair occurred

TABLE 4.2. Specification of the 6 strategies used for the
orthogonalization of vectors in a band Lanczos step.

Strategy Characteristics	1	2	3	4	5	6
First orthogonalization w.r.to older vectors (2)	Symmetry	symm.	unsymm	symm	symm	unsymm
Reort. against older (2)	no	yes	yes	no	yes	yes
Extra reort. against q_j	no	no	no	yes	yes	yes
Work/step first orthog*	$(3p+2)n$	$(5p+2)n$	$(4p+2)n$	$(3p+4)n$	$(5p+4)n$	$(4p+4)n$
+ for reort when needed	$(2p+2)n$	$(4p+2)n$	$(4p+2)n$	$(2p+2)n$	$(4p+2)n$	$(4p+2)n$

* Time for performing $y = Ax$ should be added.

Department of Information Processing
Umeå University
S—90187 Umeå, Sweden

1. D. BOLEY & G. H. GOLUB, *Inverse Eigenvalue Problems for Band Matrices*, Rep. STAN-CS-77-623, Stanford University, 1977.

2. J. CULLUM, *The Simultaneous Computation of a Few Algebraically Largest and Smallest Eigenvalues of a Large, Sparse, Symmetric Matrix*, Rep. RC6827 IBM Research, Yorktown Heights, N. Y., 1977.

3. J. CULLUM & W. E. DONATH, *A Block Generalization of the Symmetric s-Step Lanczos Algorithm*, Rep. RC4845, IBM Research, Yorktown Heights, N. Y., 1974.

4. J. W. DANIEL, W. B. GRAGG, L. KAUFMAN & G. W. STEWART, "Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization," *Math. Comp.*, v. 30, 1976, pp. 772–795.

5. G. H. GOLUB & R. UNDERWOOD, "The block Lanczos method for computing eigenvalues," *Mathematical Software* III, (J. R. Rice, Ed.), Academic Press, New York, 1977, pp. 361–377.

6. W. KAHAN & B. PARLETT, *An Analysis of Lanczos Algorithms for Symmetric Matrices*, Tech. Rep. ERL-M467, Univ. California, Berkeley, 1974.

7. C. LANCZOS, "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators," *J. Res. Nat. Bur. Standards*, v. 45, 1950, pp. 255–282.

8. J. G. LEWIS, *Algorithms for Sparse Matrix Eigenvalue Problems*, Rep. STAN-CS-77-595, Stanford, 1977.

9. C. C. PAIGE, "Practical use of the symmetric Lanczos process with re-orthogonalization," *BIT*, v. 10, 1970, pp. 183–195.

10. C. C. PAIGE, *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices*, Ph. D. Thesis, London University, 1971.

11. C. C. PAIGE, "Computational variants of the Lanczos method for the eigenproblem," *J. Inst. Math. Appl.*, v. 10, 1972, pp. 373–381.

12. B. PARLETT & D. S. SCOTT, *The Lanczos Algorithm with Implicit Deflation*, Rep. ERL M77/70, Univ. California, Berkeley, 1977.

13. A. RUHE, "Iterative eigenvalue algorithms for large symmetric matrices," *ISNM* 24, *Proc. Conf. Eigenwertprobleme Oberwolfach* 1972, Birkhauser Verlag, Basel and Stuttgart, 1974, pp. 97–115.

14. A. RUHE, "Computation of eigenvalues and eigenvectors," *Sparse Matrix Techniques Copenhagen* 1976, Lecture Notes in Math., vol. 572, Springer-Verlag, Berlin-Heidelberg-New York, 1977, pp. 130–184.

15. A. RUHE & T. WIBERG, "The method of conjugate gradients used in inverse iteration," *BIT*, v. 12, 1972, pp. 543–554.

16. R. UNDERWOOD, *An Iterative Block Lanczos Method for the Solution of Large Sparse Symmetric Eigenproblems*, Rep. STAN-CS-75-496, Stanford University, 1975.

17. J. H. WILKINSON & C. REINSCH (Eds.), *Handbook for Automatic Computation*, Vol. II: *Linear Algebra*, Springer-Verlag, Berlin-Heidelberg-New York, 1971.