Trail to a Lyapunov equation solver

Eugene L. Wachspress
49 Meadow Lakes 07, Hightstown, NJ 08520, United States

Received 18 April 2007; accepted 25 April 2007

Abstract

The Lyapunov matrix equation $AX + XA^T = B$ is $N$-stable when all eigenvalues of the real $n \times n$ matrix $A$ have positive real part. When the real $n \times n$ matrix $B$ is spd the solution $X$ is spd. It is of low rank when $B = CC^T$ where $C$ is $n \times r$ with $r \ll n$. An efficient algorithm has been found for solving the low-rank equation. This algorithm is a result of over fifty years of research starting with seemingly unrelated development of alternating direction implicit (ADI) iterative solution of elliptical systems. The low rank algorithm may be applied to a full rank equation if one can approximate the right-hand side by a sum of low rank matrices. This may be attempted with the Lanczos algorithm.

Keywords: Lyapunov equation; Low rank; ADI iteration; Matrix decomposition; Lanczos

1. Early ADI analysis

The ADI method for solving elliptical systems was introduced by Peaceman and Rachford in 1955 [1]. The five-point Laplacian difference equations yield the matrix equation $Au = s$ for which $A = H + V$, where $H$ and $V$ are the spd components from the $x$ and $y$ derivatives, respectively. The iteration equations are:

\begin{align}
(H + p_j I)u_{j-1/2} &= (p_j I - V)u_{j-1} + s \\
(V + q_j I)u_j &= (q_j I - H)u_{j-1/2} + s.
\end{align}

Iteration parameters $p_j$ and $q_j$ are chosen to optimize convergence [2]. Rigorous analysis of convergence was developed only for commuting $H$ and $V$, and associated systems were called “ADI model problems”. For such problems the method is very efficient with error reduction varying as the logarithm of the $p$-condition number of $A$. Optimum iteration parameters were determined through application of theory originally exposed by Zolotarev in 1877 [3] and applied to ADI in [4]. This Chebyshev minimax theory draws heavily on modular transformation of elliptical functions. When $H$ and $V$ do not commute, convergence can deteriorate significantly and the ADI iteration may even diverge [5]. This deficiency may be removed by preconditioning an actual problem with a related commuting one [6]. An improved preconditioning was developed twenty years later [7]. This analysis was restricted to real spectra.
2. The Lyapunov matrix equation

In 1984 it was recognized that the Lyapunov matrix equation

$$AX + XA^\top = B$$

(with eigenvalues of the real $n \times n$ matrix $A$ having positive real part and $B$ spd) is an ADI model problem [8]. This follows from the identity: $(AX)A^\top \equiv A(XA^\top)$. Pre- and postmultiplication of $X$ gives the same result when performed in either order. Application now required generalization of ADI iteration to complex spectra. The associated minimax theory is simplified through use of Rouche’s theorem [9]. Several approaches were attempted for finding good iteration parameters. The first attempts [10–14] involved embedding the actual spectrum in an “elliptical-function region”. This approach was refined in [2]. Extensive use of theory of elliptical functions was required. Concurrently, theory related to Leja and Fejér points [15] was applied by Starke [16,17]. An alternative approach based on Kolmogorov optimality analysis [18,19] was implemented by Istace and Thiran [20] with an exchange algorithm of Osborne and Watson [21]. The elliptical function region approach has been implemented with a set of MATLAB programs for solving Lyapunov, Sylvester and generalized Sylvester equations. The Sylvester equation is $AX + XB = C$, where $A$ is $n \times n$, $B$ is $m \times m$, and $X$ and $C$ are $n \times m$. Analysis now required generalization of the two-variable ADI minimax problem from real [4] to complex spectra [2].

ADI iteration applied to the actual equation is expensive because of the need to solve two linear systems of the form $(A + w_i I)u = k$ for each ADI iteration with parameters $w_i$, $s = 1, 2, \ldots, t$. In addition, the optimal parameter set depends on estimation of certain crucial values in the spectrum of $A$, this being a computation intensive task for a large full matrix $A$. One may apply a similarity transformation to the Lyapunov equation with a nonsingular matrix $G$ to yield the system $SY + YS^\top = F$, where $S = GAG^{-1}$, $F = GBG^\top$, $Y = GXG^\top$. A scheme was devised for reducing $A$ to a banded upper Hessenberg matrix $S$ by a sequence of row and column gaussian transformations [22,23]. This led to a significant reduction in computation and made ADI iteration competitive with the commonly used BS method described by Bartels and Stewart [24] and by Golub, Nash and vanLoan [25].

Although the theory underlying ADI iteration is elegant and the method has been implemented with reasonable success, the simplicity of BS continues to make this the method of choice. BS requires neither reduction to banded Hessenberg form nor determination of iteration parameters from the matrix spectrum. A method proposed by Smith [26] is also used. This is equivalent to a one parameter ADI type iteration coupled with a matrix squaring each step that effectively results in error reduction of $2^t$ normal ADI iterations after $t$ Smith iterations. The matrix squaring increases the number of nonzero elements in the coefficient matrix each step so that preliminary reduction is offset. Each squaring of a dense matrix is $O(n^3)$ so that Smith’s method is efficient only for problems requiring a small value for $t$ to be competitive with BS and ADI. Problems of large order have been solved with ADI iteration more efficiently than would have been possible with BS or Smith. The ADI method has not yet been included in any major software package. For example, BS is in the MATLAB control theory package.

3. Low rank Lyapunov equations

A low rank problem is one where $B = CC^\top$ where the $n \times r$ matrix $C$ is of rank $r$ with $r \ll n$. Although attempts have been made to modify BS to reduce computation when the right hand side is of low rank [27], only modest improvement has been obtained. It was recognized by Thilo Penzl [28] that the ADI approach could be formulated to take full advantage of a low rank $B$. (Professor Penzl’s contribution ended unfortunately with his death due to a climbing accident prior to publication of this seminal work.) When $B$ is of rank $r$, the ADI approximation to $X$ increases in rank by $r$ after each iteration. Penzl’s ADI iteration $s$ operated on a matrix of order $n \times sr$ and the approximation after $J$ steps was of rank $rJ$. This analysis established that accurate approximations to the solution of low rank problems were themselves of low rank.

Subsequently, Li and White [29] improved on Penzl’s method by reducing the ADI iteration arithmetic to operating on $r$ vectors of length $n$ each iteration. Li and White implemented their method without preliminary reduction of $A$ to tridiagonal or banded Hessenberg form. They developed procedures for efficient solution with a relatively simplistic determination of spectral parameters required to obtain ADI iteration parameters. The Li–White scheme for $J$ iterations with the full matrix $A$ is $O(rJn^2)$. If one first reduces $A$ to banded Hessenberg form, $S$, the iteration is only $O(rJn)$. However, the initial reduction and recovery of $X$ from $Y$ is $O(n^3)$. Accurate determination of the spectrum...
of $S$ with the standard MATLAB EIG program requires additional $O(n^3)$ arithmetic. This enables determination of more nearly optimal ADI iteration parameters with a subsequent reduction in $J$, but the primary gain is from the near tridiagonal system for the iteration.

Li and White give a strong argument for applying their algorithm without prior reduction. This is especially valid for real spectra. However, when $A$ is not symmetrical, efficient application of ADI iteration requires not only accurate determination of spectral bounds but fairly accurate estimation of eigenvalues near the imaginary axis. Double-implicit LR iteration applied to the banded upper Hessenberg matrix $S$ with relaxed bounds on gaussian multipliers to reduce permutations leads to much less fill-in than QR iteration. This offers the possibility of $O(n^2)$ spectrum determination. Although this approach often succeeds, it lacks the robustness of QR. The availability of the efficient and robust QR algorithm favors its use in general.

4. The Penzl algorithm

Penzl’s algorithm will now be described. The reduced equation is

$$SY + YS^\top = H,$$

(3)

where $S = GAG^{-1}$, $Y = GXG^\top$, and $H = FF^\top$ with $F = GC$ of rank $r \ll n$. Matrix $G^{-1}$, used to recover $X$ from $Y$, is accumulated during the two-sided gaussian reduction of $A$ to $S$. The nonfactored ADI iteration equations with $Y_0 = 0$ and the number of iterations $J$ determined from the spectrum and a prescribed bound on the solution error are:

$$[S + w_jI]Y_j = H + Y_{j-1}[w_jI - S]^\top$$  (4.1)

$$[S + w_jI]Y_j = H + Y_{j-1}[w_jI - S]^\top,$$  (4.2)

for $j = 1, 2, \ldots, J$. For each value of $j$, matrix $S + w_jI$ is factored and the $2n$ linear systems for the columns of $Y_{j-1/2}$ and then $Y_j$ are solved. The number of iterations $J$ is often $O(\log n)$. The reduction to Eq. (3) and recovery of $X$ from $Y$ are the $O(n^3)$ steps. Although $Y_j$ is symmetrical, $Y_{j-1/2}$ is in general not symmetrical. If the iteration matrix for the $j$-th step is defined as

$$R_j = [w_jI + S]\^{-1}[S - w_jI],$$

(5)

then

$$Y_j = 2w_j[w_jI + S]\^{-1}H[w_jI + S]^{-\top} + R_jY_{j-1}R_j^\top.$$  (6)

Eq. (6) provides the basis for improved efficiency when $r \ll n$. Let the ADI approximation after iteration $j$ with parameter $w_j$ be $Y_j$. Suppose

$$Y_j = \sum_{i=1}^{j} Z_i(j)Z_i(j)^\top.$$  (7)

Then, with $Z_i(0) = 0$ Eq. (7) is satisfied when

$$Z_j(j) = \sqrt{2w_j[w_jI + S]^{-1}F}$$

(8.1)

$$Z_i(j) = R_jZ_i(j-1), \quad i = 1, 2, \ldots, j-1.$$  (8.2)

Now the ADI iteration of Eq. (8) replaces the two steps in Eq. (4). Moreover, $Y_j$ is of rank $jr$. Note that $Z_i(j - 1)$ is a matrix of order $n \times r(j - 1)$ so the algorithm requires solving $rJ(rJ + 1)/2$ linear systems of order $n$.

A complication arises when an iteration parameter $w_j$ is complex, which may be optimal for the ADI iteration but which increases the arithmetic for the $j$-th iteration. Complex parameters always occur in conjugate pairs since $A$ and $S$ are real. It has been recommended by several researchers that the two iterations with conjugate parameters be combined to yield a coefficient matrix of the form $A^2 + pA + qI$ with real $p$ and $q$. When $A$ is reduced to upper Hessenberg form $S$, the upper bandwidth of $S^2$ and hence of $S^2 + pS + qI$ is twice that of $S$. The effort is about the same as for two iterations with real parameters.
The reduced equation form when complex parameters are used is maintained by rewriting Eqs. (4) as:

\[ [S + w_j I]Y_{j-1/2} = H + Y_{j-1}[w_j I - S^T] \]  
(9.1)

\[ Y_j[S^T + w'_j I] = H + [w'_j I - S]Y_{j-1/2}, \]  
(9.2)

where \( w' \) is the complex conjugate of \( w \). Then for this iteration, Eqs. (5) and (8) become

\[ Q_j = [w_j I + S]^{-1}[S - w'_j I], \]  
(10.1)

\[ Z_j(j) = \sqrt{w_j + w'_j[w_j I + S]^{-1}F}. \]  
(10.2)

\[ Z_i(i) = Q_j Z_i(j - 1) \quad i = 1, 2, \ldots, j - 1. \]  
(10.3)

We observe that when \( w_{j+1} = w'_j \), \( Q_j Q_{j+1} = R_j R_{j+1} \). Thus, the ADI iteration matrix is recovered when the roles of \( w \) and \( w' \) are interchanged for the iteration with the conjugate parameter. Since this introduces complex \( Z_j \) and complex arithmetic requires more flops than real arithmetic, the complex iteration parameters are saved for last. Repeated use of a single real parameter with more iterations may sometimes be more efficient than the "optimal" set of complex parameters. In some cases, it is best to apply complex parameters only to reduce error associated with eigenvalues close to the imaginary axis and to embed the remaining spectrum in a disk for which repeated use of a real parameter is optimal.

A measure of solution accuracy is found by plugging it into the equation. For the full system one computes \( AY \) and the residual error \( \|H - AY - YA^\top\|_1 \). For the low-rank system, \( W_i = SY_i \) is first computed for each \( i \) and then \( U_i = W_i Y_i^* \) is summed to yield \( SY \).

5. The Li–White algorithm

The Li–White recursive algorithm will now be described. This algorithm was qualified with a MATLAB program. Reduction to banded upper Hessenberg form was common to all Lyapunov solvers studied. By Eq. (10.2),

\[ Z_J(J) = \sqrt{w_j + w'_j[w_j I + S]^{-1}F}. \]  
(11.1)

\[ Z_{J-1}(J - 1) = \sqrt{w_{J-1} + w'_{J-1}[w_{J-1} I + S]^{-1}F}. \]  
(11.2)

By Eqs. (10) and (11),

\[ Z_{J-1}(J) = [w_j I + S]^{-1}[S - w'_j I]Z_{J-1}(J - 1) \]

\[ = \sqrt{w_{J-1} + w'_{J-1}[w_{J-1} I + S]^{-1}[S - w'_j I]Z_J(J)}. \]  
(12)

In general, proceeding back from \( i = J \) to 1, the \( r \) columns of each \( Z_i(J) \) are computed in succession:

\[ Z_J(J) = \sqrt{w_j + w'_j[w_j I + S]^{-1}F}. \]  
(13.1)

\[ Z_{i-1}(J) = \sqrt{w_{i-1} + w'_{i-1}[I - (w'_j + w_{i-1})(S + w_{i-1} I)^{-1}]Z_i(J), \quad i = J, J - 1, \ldots, 2. \]  
(13.2)

The result is independent of parameter ordering. Complex arithmetic is reduced by ordering all complex parameters ahead of real parameters since the algorithm starts with \( w_j \) and proceeds backward. Now each iteration requires solution of only \( r \) linear systems for a total of \( rJ \) systems rather than the \( rJ(rJ + 1)/2 \) required of the Penzl algorithm.

6. Sylvester equations

This approach also applies to Sylvester equations. Consider the banded system of order \( n \times m \):

\[ SY + YT = EF^\top. \]  
(14)
of the Sylvester equation $AX + XB = H$ where $H = CD\top$. $E$ is order $n \times r$ and $F$ is order $m \times r$. The transformation matrices $L_s$ and $L_t$ are saved for computing $X = L_s Y L_t$. The ADI approximation to the solution after $j$ iterations is

$$Y_j = \sum_{i=1}^{j} U_i(j) V_i(j).$$  \hfill (15)

Now matrices $U_i(j)$ and $V_i(j)$ must be computed for each $j$. There are in general two iteration parameters, $u_j$ and $v_j$, for each iteration $j$. Matrix $R_j$ of Eq. (5) is now

$$R_j = [u_j I + S]^{-1} [v_j I - S],$$  \hfill (16)

and similarly

$$Q_j = [v_j I + T]^{-1} [u_j I - T].$$  \hfill (17)

Now Eqs. (4) become

$$[S + u_j I] Y_{j-1/2} = E F\top + Y_{j-1} [u_j I - T]$$  \hfill (18.1)

$$Y_j [T + v_j I] = E F\top + [v_j I - S] Y_{j-1/2}.$$  \hfill (18.2)

The recursion formulas for the $U_i(j)$ are

$$U_i(j) = R_j U_i(j - 1), \quad i = 1, 2, \ldots, j - 1,$$  \hfill (19.1)

$$U_j(j) = (u_j + v_j) [u_j I + S]^{-1} E,$$  \hfill (19.2)

and

$$V_i(j) = Q_j^T V_i(j - 1), \quad i = 1, 2, \ldots, j - 1,$$  \hfill (20.1)

$$V_j(j) = [v_j I + T]^{-1} F.$$  \hfill (20.2)

The Li–White algorithm may be introduced to reduce iteration complexity.

7. Approximation of the right-hand side by a sum of low-rank matrices

The analysis applies when the right-hand side of the reduced Lyapunov equation (Eq. (3)) is of the form

$$H = F P F\top$$

with $P$ of order $r \times r$ or the r.h.s. of the reduced Sylvester equation (Eq. (14)) is $E P F\top$. The matrix $P$ does not affect the ADI iteration equations. Significant reduction in computation may be realized if the r.h.s. can be approximated reasonably well in this form. A review of matrix factorization with discussion of low-rank approximations is given in [30]. Penzl observed that “splitting up the right-hand side matrix into a sum of low-rank matrices enables an efficient parallelization of [his] method”.

This suggests a general procedure for solving all $N$-stable Lyapunov problems. Let the $n \times m$ matrix of orthonormal vectors of $m$ Lanczos steps applied to matrix $H$ be $K$ and let the Lanczos coefficients determine the tridiagonal matrix $T$ of order $m$. Then the matrix $F_j = K T K\top$ is a rank $m$ approximation to $H$. We may, therefore, compute

$$H_1 = H - F_1$$

and perform $m$ Lanczos steps on $H_1$ with initial vector equal to what would have been the $m + 1$ vector of the previous Lanczos steps on $H$. This may be continued to yield a set of $F_j$. The norms of successive $H_j$ should decrease and the algorithm may be terminated when sufficient accuracy is achieved with the sum of the low-rank approximations. If $H$ is of full rank with $n = 100$ and $m$ is 5, for example, the algorithm should terminate when $j$ is around 20. The matrix $H_{21}$ will in general not be the zero matrix since the Lanczos vectors from each $H_j$ are not orthogonal to the previous vectors. When the sum of the low rank subspaces approaches $H$, the rank of subsequent subspaces may be smaller than $m$.

In the MATLAB research program LIWHITE07, when the absolute value of the $(k, k+1)$ element of $T$ (for $k < m$) is less than .001 times that of element $(1, 2)$ the rank of the subspace is chosen as $k$. The algorithm is terminated when the order of $T$ is one. Having generated a set of low rank matrices, one may solve the low rank Lyapunov systems in parallel. In this application, the matrix $A$ and its transformation to $S$ is common to all low-rank problems. The
reduction, spectrum evaluation, ADI iteration parameter determination and back transformation need only be done once. This approach extends application of the Li–White algorithm to general right-hand sides. When \( H \) is not given in factored form but is of rank \( r \ll n \) the Lanczos partitioning will expose the rank deficiency of \( H \) and may lead to more efficient solution.

In LIWHITE07 the ADI rhs for each low-rank subspace is \( K \) and the ADI result is \( Y \). The contribution to the solution is \( ZY = YTY^\top \). The sum of the subspace results is \( Z_{\text{tot}} \). The matrix \( B \) in Eq. (2) was chosen as \( A + A' \) for a set of test problems. This \( B \) is symmetrical but not necessarily spd. In general, the unique solution \( X \) need not be spd when \( B \) is not spd. However, this choice for \( B \) yields the solution \( X \) equal to the identity matrix. For the low-rank equations, it should be noted that the tridiagonal matrices \( T \) need not be spd since only \( K \) appears as a rhs for the ADI iterations.

One test case was run with a random \( N \)-stable \( A \) of order 30 for which \( B \) (and hence the rhs \( H \) of the transformed equation) was not spd. The initial value for \( m \) was chosen as 5. The first seven subspaces were of rank 5, but the eighth was of rank 1. The sum of the low-rank solutions agreed with the true solution to four significant places. Another problem was solved with \( A \) of order 100 and \( m = 10 \). The first eleven subspaces were of rank 10 and the twelfth was of rank 1. The factored \( Y \) agreed with the non-factored \( Y \) to four significant figures. Comparable accuracy was obtained with 14 subspaces of rank \( m = 8 \) and a fifteenth of rank 1. For this problem, an initial choice of \( m = 12 \) resulted in nine subspaces of rank 12, a tenth of rank 5 and a last subspace of rank 1. The factored result agreed with the nonfactored result to five significant places. To illustrate how the Lanczos algorithm exposes rank deficiency of a given full matrix, a random full matrix of order 100 and rank 20 was chosen as \( B \). The value for \( m \) was chosen as five. The algorithm terminated with four subspaces of rank 5, one of rank 4 and the last of rank 1.

The program RELANC refines a LIWHITE07 result \( X_1 \) with \( H \) approximated by \( W = \sum H_j \) when \( \text{Err}(X_1) \equiv \|B - AX_1 - X_1A^\top\|/\|B\| \) exceeds a specified bound. One computes the solution \( X_2 \) to the reduced problem with right-hand side \( H^* = H - W \) and computes \( X = X_1 + X_2 \) as the refined approximation. A test problem of order \( n = 100 \) gave \( \text{Err}(X_1) = .0022 \) and \( \text{Err}(X_1 + X_2) = 3.8e - 9 \). For this problem the initial partition of \( H \) was into eleven matrices of rank 10 and one of rank 1. The RELANC partition was into four matrices of rank 10, one of rank 5 and a fifth of rank 4.

An in-depth comparison of computation time for various approaches should be made. Many stages are easily parallelized. Reduction to banded upper Hessenberg form, the Lanczos algorithm, GMRES type solution of the ADI iteration equations with sparse \( A \), and simultaneous solution for all low-rank matrices generated by the Lanczos algorithm applied to a full-rank system are among these stages. It should be noted that the ADI iteration equations may be solved for all columns of \( X \) or \( Y \) simultaneously once the right-hand sides of the equations are computed. However, computation of these right-hand sides for a full rank \( H \) of order \( n \) requires a factor of \( n/m \) times computation with the \( n \times m \) factor \( K \) in the Li–White algorithm. When the coefficient matrix \( A \) is sparse, solution without reduction may be best, provided one can determine good ADI iteration parameters. However, when \( A \) is not sparse the GMRES approach may become less efficient even with good ADI parameters. Optimization and comparison of relative efficiency of the various methods is a fertile area for further research.

8. Summary

The ADI iteration originally proposed by Peaceman and Rachford in 1955 for numerical solution of difference equations for elliptic partial differential equations has been widely used for solving such problems. Analysis is less precise when the coefficient matrix is split into noncommuting components for the iteration. Almost thirty years after inception of ADI iteration, it was recognized that Lyapunov and Sylvester matrix equations have the commutation property. This led to generalization of the ADI iteration theory from real to complex spectra with a rather elegant application of the theory of modular transformations of elliptic functions. It also stimulated research into similarity reduction of a full nonsymmetrical real matrix to sparse form, and in particular to banded upper Hessenberg form by means of row and column gaussian transformations. Despite the rapid convergence of ADI iteration for these problems, earlier methods of Bartels–Stewart and Smith remained competitive and were already incorporated in major software packages. After another twenty years had elapsed, it was observed by Penzl that low-rank equations could be treated by a low-rank ADI iteration more efficiently than by conventional methods which have not been shown to admit significant gains in efficiency for low-rank problems. The subsequent contribution by Li and White further reduced computational effort so that ADI iteration is clearly superior to the other methods for such problems. Relative merits
of iterating with the full system and iterating after reduction to banded Hessenberg form are still under consideration and may well depend on specific applications.

The possibility of approximating a general right-hand side by a sum of low-rank matrices extends application of the Li–White algorithm to all $N$-stable Lyapunov systems. The Lanczos algorithm has worked well for generating a sum of low-rank approximations in the few test problems thus far considered. In general, ADI solution of Lyapunov equations by any of the methods discussed is well suited for parallel computation.

References

