JOURNAL OF COMPUTATIONAL PHYSICS 108, 8–14 (1993)

Computation of Selected Eigenvalues of Generalized Eigenvalue Problems

NARINDER NAYAR

Deneb Robotics, P.O. Box 214687, Auburn Hills, Michigan 48321

AND

JAMES M. ORTEGA

Department of Computer Science, University of Virginia, Charlottesville, Virginia 22903

Received July 27, 1992

We examine and develop techniques for obtaining a few selected eigenvalues of the generalized eigenvalue problem $Ax = \lambda Bx$, where A and B are $n \times n$, nonsymmetric, banded complex matrices. One way of obtaining the desired eigenvalues is to use a direct method to compute all the eigenvalues. Direct methods are computationally intensive and destroy the sparsity of the matrices A and B. Iterative methods, on the other hand, maintain the sparsity of the matrices and compute only a few eigenvalues. The iterative algorithms that we consider are the Arnoldi and the Lanczos methods. We use a shift and invert strategy to increase the rate of convergence towards the desired eigenvalues. We compare these two approaches for a model problem, which arises from considering the linear stability of compressible boundary layers and some other test problems. We present a general scheme to compute the eigenvalues lying inside a "box" in the complex plane. We also outline a procedure to separate the converged eigenvalues from spurious approximations. In addition, this procedure can also improve the approximations to the eigenvalues of interest. Numerical results obtained on a CRAY Y-MP are presented.

1. INTRODUCTION

Malik [12] considered the linear stability of compressible boundary layers. The computational problem is to ascertain the eigenvalue of a generalized eigenvalue problem

$$A\mathbf{x} = \lambda B\mathbf{x} \tag{1.1}$$

that lies in a specified region in the complex plane and has largest imaginary part.

To derive (1.1), Malik [12] reduces the Navier–Stokes equations to a system of linear ordinary differential equations

$$\left(E\frac{d^2}{dy^2} + F\frac{d}{dy} + G\right)\phi = 0, \tag{1.2}$$

where, in three space dimensions, E, F, and G are 5×5 matrices and ϕ is the vector of the mean values of perturbed velocities $\tilde{u}$, $\tilde{v}$, and $\tilde{w}$, pressure $\tilde{p}$, and temperature $\tilde{T}$. The equations are discretized by finite differences to obtain (1.1). If $N+1$ is the number of grid points, the matrices A and B are $5N \times 5N$ ($4N \times 4N$ in only two space variables, as used in our numerical experiments) banded, complex, and non-Hermitian. B is nonsingular. Further details on the derivation of (1.1) and the stability problem can be found in [12]. Related results are given in [3].

In order to determine the eigenvalue controlling stability, Malik [12] computes all the eigenvalues of (1.1). Our concern in the paper is to develop alternative methods that compute the eigenvalues of interest given a box in the complex plane in which they lie. The main method that we discuss is based on Arnoldi's iteration [2, 19] and a shift and invert strategy [5, 7]. We also consider Lanczos-type methods [8, 11] but, at least for our test problems, they are slightly inferior to the Arnoldi methods. We test these methods on the stability problem as well as some other artificial problems in which several eigenvalues are required. All experiments are run on a single processor of a CRAY Y-MP.

Before considering these iterative methods, we first show in Section 2 how to improve on the approach of [12] by using LAPACK [1], the successor to LINPACK and EISPACK.

2. QR-TYPE METHODS

The approach discussed in [12] was to form $B^{-1}A$ using LU decomposition, followed by forward and back solves; use EISPACK [9] to reduce $B^{-1}A$ to Hessenberg form H; then find all the eigenvalues of H by LR iteration. Although

0021-9991/93 $5.00

TABLE I

Time (Seconds) for Different Methods

N	$B^{-1}A$	Hessenberg reduction	Eigenvalues	Total time
		Original code using LR		
16	0.047	0.025	0.031	0.103
40	0.286	0.333	0.329	0.948
60	0.641	0.962	0.939	2.541
		LAPACK and QR		
16	0.011	0.016	0.082	0.109
40	0.061	0.205	0.724	0.990
60	0.134	0.694	1.976	2.804
		LAPACK and LR		
16	0.011	0.016	0.031	0.058
40	0.061	0.205	0.329	0.595
60	0.134	0.694	0.939	1.767

A and B are banded, $B^{-1}A$ is, in general, a full matrix. An alternative to forming $B^{-1}A$ is the QZ algorithm [14], but it also does not preserve the bandwidth so we have not considered it for this problem.

We now modify the approach in [12] by replacing EISPACK with LAPACK [1]. LAPACK is based on "blocked" matrices but the block sizes for our banded matrices are so small that no benefits are achieved by using blocked code to obtain $B^{-1}A$. But since $B^{-1}A$ is a full matrix, its reduction to Hessenberg form can be done efficiently by blocking. LAPACK is then used to compute the eigenvalues of the reduced matrix, but the best results are achieved for block size equal to one. (The QR implementation in LAPACK at the time we used it was based on the block-multishift method, but this may change in later versions.) Table I compares timings on a single processor of a CRAY Y-MP for the original code of [12] and the LAPACK modification. This is for a two-dimensional problem so that the sizes of the matrices are $4N$. Although the times for forming $B^{-1}A$ and the Hessenberg reduction are reduced considerably by using LAPACK, the eigenvalue computation times increase. This is due to the fact that the QR iteration is roughly twice as slow as LR. The last part of the table shows the times if QR is replaced by the original LR code.

There is a further improvement that can be made in computing $B^{-1}A$. A and B are not only banded but have a good deal of additional structure as shown in Fig. 1. In particular, it is possible to find a permutation matrix P so that $P^{\mathrm{T}}BP = \mathrm{diag}(B_1, B_2)$, where B_2 is diagonal and the semi-bandwidth of B_1 is 3, which is less than the original B. This same reordering must be applied to A, which destroys its block tridiagonal form, but the time to compute $B^{-1}A$ is still almost halved. This reduces the total times in Table I by about 3%.

3. ARNOLDI'S METHOD

The Arnoldi algorithm [2, 19] for the standard eigenvalue problem

$$C\mathbf{u} = \lambda \mathbf{u}, \tag{3.1}$$

starts with an initial vector $\mathbf{v}_1$ with $\|\mathbf{v}_1\|_2 = 1$ and computes additional vectors $\mathbf{v}_2, \ldots, \mathbf{v}_m$ by

$$\begin{aligned} \hat{\mathbf{v}}_{j+1} &= C\mathbf{v}_j - \sum_{i=1}^{j} h_{ij}\mathbf{v}_i \\ h_{j+1,j} &= \|\hat{\mathbf{v}}_{j+1}\|_2 \\ \mathbf{v}_{j+1} &= \hat{\mathbf{v}}_{j+1}/h_{j+1,j} \end{aligned} \tag{3.2}$$

for $j = 1, \ldots, m-1$. The h_{ij} in (3.2) are the inner products

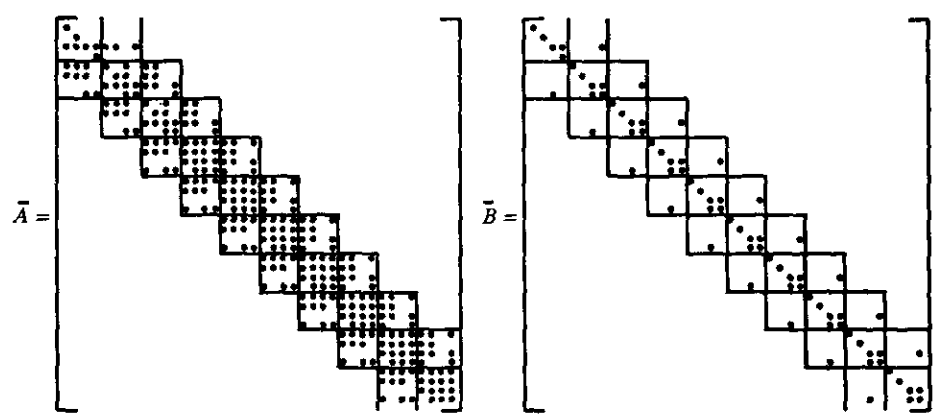

FIG. 1. Structure of matrices A and B for two-dimensional flow.

TABLE II

Arnoldi for $N = 60$, Shift $= 0.03975 + i0.05$

	Unstable eigenvalue $= 0.029118 + i0.002174$	
Steps	Real part	Imaginary part
8	0.029204	0.002006
10	0.029061	0.002255
13	0.029121	0.002174
16	0.029118	0.002174

$h_{ij} = (\mathbf{v}_i, C\mathbf{v}_j)$ so that $\mathbf{v}_{j+1}$ is orthogonal to all previous $\mathbf{v}_i$. The h_{ij} define an $m \times m$ Hessenberg matrix H whose eigenvalues approximate those of C, especially those in the outer part of the spectrum of C. Thus, one wishes to choose m sufficiently small so that the work in generating H and computing its eigenvalues by the QR method is not excessive, but m needs to be sufficiently large so that selected eigenvalues of C are approximated accurately.

If one wishes to approximate eigenvalues of C that are not in the outer part of the spectrum, it is common to use the "shift and invert" strategy [5, 7]. If λ_0 is an approximation to an eigenvalue of interest, then the shifted and inverted problem is

$$(C - \lambda_0 I)^{-1}\mathbf{u} = \mu\mathbf{u}, \tag{3.3}$$

where $\mu = 1/(\lambda - \lambda_0)$. Thus, eigenvalues of C close to λ_0 correspond to eigenvalues μ of (3.3) with large absolute value, and one expects Arnoldi's method to converge to such eigenvalues.

In order to apply Arnoldi's method to (3.3) for the generalized eigenvalue problem (1.1), we do not wish to form $C = B^{-1}A$. Rather, (3.3) may be written as

$$(A - \lambda_0 B)^{-1} B\mathbf{u} = \mu\mathbf{u} \tag{3.4}$$

and to apply Arnoldi's method we do an LU decomposition of $A - \lambda_0 B$ once, and then each time $(A - \lambda_0 B)^{-1} B\mathbf{v}$ is needed, we solve $(A - \lambda_0 B)\mathbf{w} = B\mathbf{v}$ by forward and back solves. This is much more economical than forming the matrix of (3.4) explicitly since it is usually full and also its dimension is much larger than m.

TABLE III

Times (Seconds) for $N = 60$

m	15	20	25	30	35
Arnoldi	0.046	0.058	0.075	0.087	0.099
Eigenvalues	0.009	0.017	0.024	0.035	0.050
Cumulative	0.055	0.084	0.125	0.172	0.234

If we wish to obtain eigenvalues of C in some box in the complex plane and we have no further information about the eigenvalues, then it is natural to use the center of the box as the shift λ_0 in (3.3). We now apply this strategy to the stability problem. For this problem, Malik [13] has suggested the box whose sides have lengths of approximately 0.1 and 0.12 and whose center is $\lambda_0 = 0.03975 + i0.05$. Table II shows the results of taking different numbers of steps, m, in Arnoldi's method applied to (3.4) for the problem of the previous section with $N = 60$. The eigenvalue of interest, as computed in the previous section, is shown at the top of the table.

After eight Arnoldi steps, the approximated eigenvalue has almost three digits of accuracy in the real part and two in the imaginary part. After 16 steps, the eigenvalue has been approximated to the full accuracy shown. The time required for 16 steps is approximately 0.06 s, a factor of almost 30 less than that shown in Table I. However, there are two serious and interconnected problems with this approach: How do we choose the number of Arnoldi steps and how can we be sure we have computed the correct eigenvalue?

Since we do not know the proper number of Arnoldi steps, m, in advance, we could start with a small m, do Arnoldi's method, increase m by some number, do additional Arnoldi steps, and so on, until the eigenvalue of interest converges. The extra work involved is the computation of all eigenvalues of the $m \times m$ Hessenberg matrix at each break point; this is not excessive until m becomes large. For example, in Table III we give times for 15 up to 35 Arnoldi steps for the $N = 60$ problem, as well as the times to compute the eigenvalues of the corresponding Hessenberg matrices. In the third row, we give the cumulative times under the assumption that we start with $m = 15$ and then increase m by five at each breakpoint. Thus, if we knew in advance that $m = 35$ was the proper number of steps, the time would be $0.099 + 0.050 = 0.149$, but having to obtain the eigenvalues of all the small Hessenberg matrices adds 0.085. In any case, this is still much faster than computing all the eigenvalues, as in the previous section.

4. LANCZOS' METHOD

We next consider the Lanczos algorithm [11] as an alternative to Arnoldi's method. Here, one generates two sets of vectors $\mathbf{v}_i$ and $\mathbf{w}_i$ that satisfy the recursions (for the problem (3.1))

$$C\mathbf{v}_j = \gamma_{j-1}\mathbf{v}_{j-1} + \alpha_j\mathbf{v}_j + \beta_j\mathbf{v}_{j+1}, \qquad \gamma_0\mathbf{v}_0 \equiv 0, \tag{4.1}$$

$$\mathbf{w}_j^* C = \beta_{j-1}\mathbf{w}_{j-1}^* + \alpha_j\mathbf{w}_j^* + \gamma_j\mathbf{w}_{j+1}^*, \qquad \beta_0\mathbf{w}_0^* \equiv 0, \tag{4.2}$$

TABLE IV

Lanczos for $N = 60$, Shift $= 0.03975 + i0.05$

	Unstable eigenvalue $= 0.029118 + i0.002174$	
Steps	Real part	Imaginary part
7	0.029635	0.001799
8	0.029139	0.002180
9	0.029124	0.002194
10	0.029119	0.002174
11	0.029118	0.002174

for $j = 1, ..., m$. The α, β, and γ determine the tridiagonal matrix

$$T = \begin{bmatrix} \alpha_1 & \gamma_1 & & & \\ \beta_1 & \alpha_2 & \gamma_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \gamma_{m-1} \\ & & & \beta_{m-1} & \alpha_m \end{bmatrix} \tag{4.3}$$

whose eigenvalues approximate the outermost eigenvalues of C. The γ_i and β_i in (4.3) are not uniquely determined and there are several possibilities. Two are $\beta_j = 1$ and $\beta_j = \gamma_j$. Our reported experiments use the first choice, although the second gives similar results.

A zero β_j terminates (4.1) but the eigenvalues of T_j are then eigenvalues of C. However, a zero γ_j is a problem that can sometimes be dealt with by the "look-ahead" Lanczos procedure [8, 18]. There are also problems in loss of orthogonality; the vectors $\mathbf{v}_i$ and $\mathbf{w}_i$ are biorthogonal, $\mathbf{v}_j^* \mathbf{w}_i = 0$, $i \neq j$, but rounding error destroys this property. This can lead to loss of accuracy in the computed eigenvalues as well as the appearance of "spurious" eigenvalues.

We now apply the Lanczos algorithm to the shifted and inverted problem (3.4). As with Arnoldi's method, we do not form the matrix explicitly; $A - \lambda_0 B$ is LU decomposed and these factors are used to compute the matrix–vector product $(A - \lambda_0 B)^{-1} B\mathbf{v}$. Results for the problem of Table II are given in Table IV for the Lanczos algorithm.

TABLE V

Case I: Shift $= 0.15 - i0.2$

Eigenvalue #	1	2	3	4	5	6	7
Number of steps	13	22	18	13	14	21	28

Table IV shows that we get full accuracy in 11 Lanczos steps as opposed to 16 for Arnoldi. However, the time required by Lanczos is approximately 0.08 s as opposed to 0.06 for Arnoldi. This is due to the fact that each Lanczos iteration requires two forward and back solves as opposed to only one for Arnoldi. Thus, on this problem, Arnoldi is slightly more efficient.

5. ADDITIONAL TEST PROBLEMS

We next consider additional problems in which more than a single eigenvalue in a specified region is desired. These are artificial problems designed to test our algorithms. We would like the matrices A and B to again be banded and we wish to generate A and B so that the problem (1.1) has known eigenvalues. We do this by first computing a banded block upper triangular matrix as the Kronecker product C of a 4×4 circulant matrix (see, e.g., [16]) and a banded upper triangular matrix. The eigenvalues of a circulant matrix are easily computed and, thus, the eigenvalues of C are known. We next choose any banded complex matrix B and define the banded matrix $A = BC$. Then $C = B^{-1}A$ has the same eigenvalues as (1.1).

By the above approach, we generate banded matrices A and B of size 100 and consider rectangles in the complex plane as illustrated in Fig. 2. In Case I, the rectangle contains seven eigenvalues, whereas in Case II, it contains nine. In Case I, the center of the rectangle should be an adequate shift, but this is not true in Case II since the rectangle is so flat. Hence, we subdivide it into four squares and we will use the center of each square as a shift.

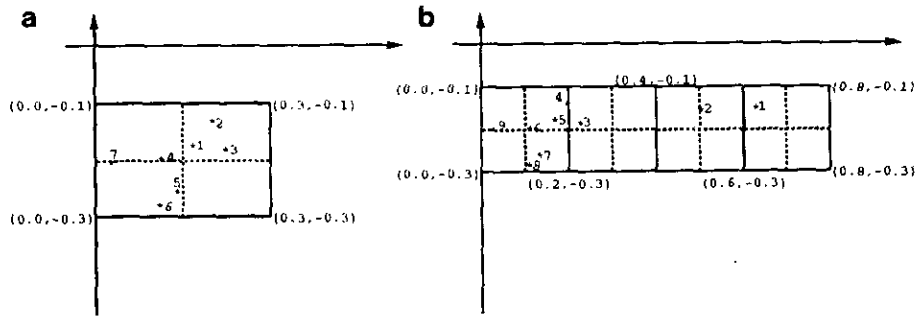

FIG. 2. Test cases: (a) Case I; (b) Case II.

TABLE VI

Case II

Shift	Eigenvalue #	1	2	3	4	5	6	7	8	9
$0.1 - i0.2$	Number of steps	—	—	29	26	21	6	16	23	18
$0.3 - i0.2$	Number of steps	34	27	15	24	25	—	31	—	—
$0.5 - i0.2$	Number of steps	12	10	—	—	—	—	—	—	—
$0.7 - i0.2$	Number of steps	10	17	—	—	—	—	—	—	—

Table V shows the number of Arnoldi steps required to obtain each eigenvalue in the rectangle to six-digit accuracy. As expected, the number of steps is roughly proportional to the distance of the eigenvalue from the shift.

Table VI shows corresponding results for Case II using the centers of the four small squares as shifts. The predominant concentration of the eigenvalues is in square A, whose center is $0.1 - i0.2$, and all the eigenvalues except 1 and 2 are computed in no more than 30 steps. The centers of the other squares are best for the eigenvalues closest to the shift points, as expected.

The Lanczos algorithm was also applied to these two cases. As with the previous stability problem, the Lanczos algorithm tended to require fewer steps than Arnoldi but a longer time.

6. A FILTERING PROCEDURE

We earlier mentioned the problem with the Lanczos algorithm regarding the generation of spurious eigenvalues, and we need a mechanism to separate "good" eigenvalues from "bad" ones. The "bad" eigenvalues include undesired eigenvalues outside the user defined box as well as spurious ones. "Good" eigenvalues are ones that have converged within the box as well as ones that have not yet converged. We would like to improve on the latter.

We consider a slightly enlarged box to account for any approximations that are converging to desired eigenvalues but are not yet themselves in the box. This serves as the first test to separate "good" eigenvalues from "bad" ones. We then define the following additional tests:

(1) *Quotient test.* This test is based on an a posteriori analysis for a computed eigenvalue (see Wilkinson [20]). If the condition number of an eigenvalue is not too large, then it is expected that the generalized Rayleigh quotient gives a good approximation to the eigenvalue in question. For the standard eigenvalue problem $C\mathbf{x} = \lambda\mathbf{x}$, the left eigenvector is defined as $\mathbf{y}^*C = \lambda\mathbf{y}^*$, the condition number as $\|\mathbf{y}\|_2 \|\mathbf{x}\|_2 / |\mathbf{y}^*\mathbf{x}|$, and the generalized Rayleigh quotient as $\mathbf{y}^*C\mathbf{x}/\mathbf{y}^*\mathbf{x}$.

The left and the right eigenvectors corresponding to λ are approximated by solving $(A - \lambda_0 B)\mathbf{r} = \mathbf{d}$ and

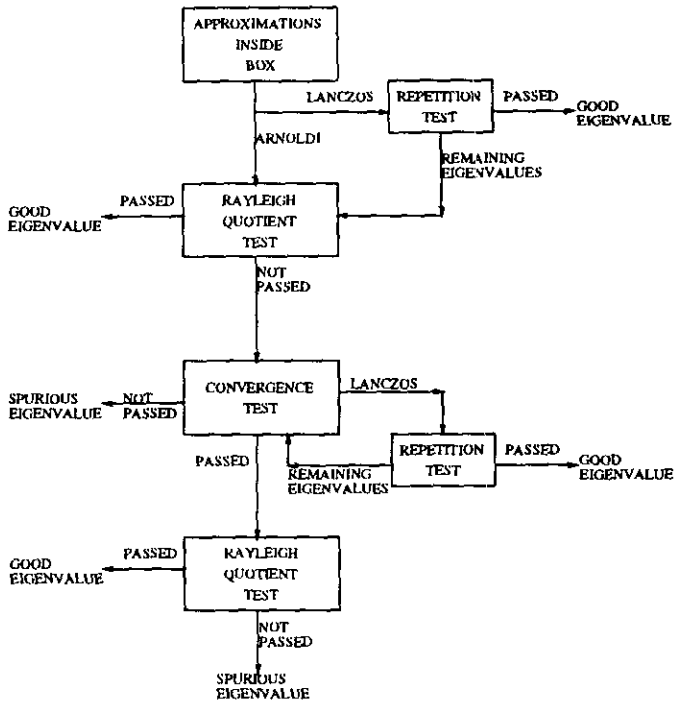

FIG. 3. Filtering scheme.

$(\bar{A}^* - \lambda_0^* \bar{B}^*)\mathbf{l} = \mathbf{d}$, where $\mathbf{d}$ is chosen as outlined in Golub and Ortega [10, p. 237]. We then transform these eigenvector approximations to the corresponding ones for the standard eigenvalue problem. If the current approximate eigenvalue and the generalized Rayleigh quotient are equal up to a tolerance, we output the condition number and the approximation.

(2) *Convergence test.* If we do not have a sufficiently good approximation to the desired eigenvalue, the quotient test will not be passed since eigenvectors are typically more sensitive than eigenvalues. To improve the approximation as well as detect spurious eigenvalues, the convergence test uses the approximation as a shift for a few more steps, say,

TABLE VII

Filtering Results Using Arnoldi's Algorithm for Case I

Steps	Eigenvalues filtered by: Quotient test	Eigenvalues filtered by: Convergence and quotient tests
9	—	1, 4, 5
13	1, 4	3, 5
14	1, 4, 5	3
16	1, 4, 5	3, 6
17	1, 4, 5	2, 3, 6
18	1, 3, 4, 5	2, 6
20	1, 3, 4, 5, 6	2, 7

5. The starting vector is obtained by solving $Uy = z$, where z is a vector of all ones and U is from the LU decomposition of $A - \lambda_0 B$. (Perhaps using the LU decomposition $(A - \lambda_0 B)\, B^{-1}$ would provide a better starting vector.) If the shift is close to any of the five newly obtained approximations, up to a tolerance, then it is classified as a possibly "good" eigenvalue. However, if none of the approximations are close (up to a tolerance), then the shift is labeled as a spurious eigenvalue. For the Lanczos procedure, the only approximations (from the five new ones) checked for proximity to the shift are those that do not pass the Repetition test. The possibly "good" eigenvalue still has to pass the quotient test to be certified as "good."

(3) *Repetition test.* For the Lanczos algorithm, Paige [17] has shown that numerically multiple copies of an eigenvalue imply convergence of that eigenvalue. Thus, duplicated approximations are accepted as "good" eigenvalues if they are inside the box of interest.

The order in which these tests are applied is shown in the schematic in Fig. 3. Further details are given in Nayar [15].

We illustrate the above "filtering scheme" with Case I of Fig. 2. Results are shown in Table VII. After nine steps of Arnoldi's algorithm, seven approximations are inside the box of interest. The quotient test does not indicate convergence for any of these, but the convergence test and, subsequently, the quotient test identifies three approximations as "good" eigenvalues. These three are the ones closest to the original shift. After 14 steps, the three eigenvalues nearest to the shift are obtained by the quotient test, and another by the combination of the convergence and quotient tests. All seven desired eigenvalues are obtained after 20 steps, five by the quotient test and the remaining two by the convergence and quotient tests.

We note that the convergence test can dramatically improve the accuracy of an approximation. For example, after nine steps, there are three approximations that are moving towards eigenvalues of interest but have only about four digits of accuracy. The convergence test applied to these approximations gives new approximations that have almost 12 digits of accuracy. However, the convergence test is fairly expensive to apply. Furthermore, we do not really need to achieve such a high degree of accuracy. In most cases, we would like the approximations to pass the quotient test directly since that requires relatively less computation.

Table VIII shows corresponding results for the Lanczos algorithm, again for Case I. Two eigenvalues are obtained after only six steps by the convergence test and, subsequently, the quotient test. After 10 steps, we obtain seven approximations inside the box of interest. Three of these pass the quotient test and the remaining ones are identified as "bad" eigenvalues. After 12 steps, we still have seven approximations inside the region of interest. However, the accuracy has improved enough so that we are able to identify five eigenvalues, three by the quotient test and the others by the combination of the convergence and quotient tests. All seven eigenvalues are obtained after 15 steps, one by using the repetition test on the five approximations obtained during the convergence test, another by the convergence test, and the remaining ones by the quotient test.

We next show in Table IX some timing results for Arnoldi's method for Case I. The "Arnoldi time" is that time required to do m steps of the algorithm and compute the eigenvalues of the $m \times m$ Hessenberg matrix.

Table IX begins with 20 steps, the smallest number for which all the desired eigenvalues are obtained. As the number of steps increases, the Arnoldi time increases but the

TABLE VIII

Filtering Results Using Lanczos Algorithm for Case I

Steps	Eigenvalues filtered by: Repetition test	Quotient test	Convergence and repetition tests	Convergence and quotient tests
6	—	—	—	1, 4
8	—	—	4	1, 5
9	—	1, 4	5	—
10	—	1, 4, 5	—	—
12	—	1, 4, 5	—	3, 6
13	—	1, 4, 5	3	6
15	—	1, 3, 4, 5, 6	7	2

TABLE IX

Arnoldi Timings (Seconds) for Case I

Steps	Arnoldi time	Filtering time	Total time
20	0.037	0.089	0.126
25	0.053	0.077	0.130
28	0.060	0.057	0.117
30	0.066	0.057	0.123
35	0.086	0.057	0.143
40	0.108	0.057	0.165

TABLE X

Lanczos Timings (Seconds) for Case I

Steps	Lanczos time	Filtering time	Total time
15	0.043	0.110	0.153
20	0.059	0.090	0.149
25	0.076	0.053	0.129
30	0.097	0.048	0.145
35	0.116	0.066	0.182
40	0.146	0.080	0.226

filtering time decreases to a constant 0.057 s. The minimum total time occurs for 28 steps. The decrease in the filtering time is due to the fact that, for 20 steps, some of the eigenvalues are obtained by the quotient test and the remaining ones by the combination of the convergence and the quotient tests (see Table VII). For a larger number of steps, more eigenvalues are obtained by the quotient test, which is less expensive than the convergence test. For 28 or more steps, all eigenvalues are obtained by the quotient test and the filtering time remains constant thereafter.

Corresponding results for the Lanczos algorithm are shown in Table X and are more variable due to the unpredictable behavior of the repetition test. For 15 steps, five of the eigenvalues are obtained by the quotient test, another one by the convergence and quotient tests, and the last by the repetition test on the five new approximations obtained during the convergence test. For 20 steps, all the eigenvalues are obtained by the quotient test. For 25 steps, numerical multiplicity occurs in the approximations obtained using the center of the rectangle as the shift. Then, two eigenvalues are sorted by the repetition test while the remaining ones are obtained by the quotient test. The lesser use of the quotient test gives the minimum total. For more than 28 steps, the filtering time starts increasing because of numerical multiplicity for eigenvalues other than those nearest to the shift.

The best Arnoldi time for Case I, 0.117 s for 28 steps in Table IX, is slightly better than the best Lanczos time of 0.129 s for 25 steps in Table X. Due to the shift and invert strategy, two forward and back solves are required for each Lanczos iteration, as opposed to only one for the Arnoldi iteration. This is reflected in Tables IX and X, where the Lanczos time is consistently higher than Arnoldi's algorithm for the same number of steps. The unpredictability of the filtering results makes it hard to compare the computational complexity of the two algorithms. However, for the original stability problem as well as other test cases, the timing results for Arnoldi's algorithm are better than for the Lanczos algorithm. Further numerical experiments in Nayar [15] show that the iterative method times including filtering are far superior to the direct method for larger problem sizes.

7. CONCLUSIONS

We have shown that the computation of one or a few eigenvalues in a specified region in the complex plane is efficiently done by use of iterative methods with a shift and invert strategy. Experiments on a limited number of problems showed that the Arnoldi iteration was somewhat faster than the Lanczos method, although the latter used fewer iterations. This conclusion also held after the addition of a "filtering" test.

ACKNOWLEDGMENTS

The work of the first author was partly supported by NASA Grant NAG-1-242 and that of the second author by NASA Grant NAG-1-1050. The authors thank Dr. T. Zang of NASA Langley Research Center for suggesting this research and Dr. M. Malik of High Technology Corporation for several helpful discussions and the use of his codes. We are also grateful to the referees for suggestions that improved the paper.

REFERENCES

1. E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. DuCroz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK User's Guide* (SIAM, Philadelphia, 1992).
2. W. Arnoldi, *Q. Appl. Math.* **9**, 17 (1951).
3. K. Christodoulou and L. Scriven, *J. Sci. Comput.* **3**, 355 (1988).
4. J. Cullum and R. Willoughby, "A Practical Procedure for Computing Eigenvalues of Large Sparse Nonsymmetric Systems," in *Large-Scale Eigenvalue Problems*, edited by J. Cullum and R. Willoughby (Elsevier Science, North-Holland, Amsterdam, 1986), p. 193.
5. J. Cullum and R. Willoughby, Research Report RC 15835 (#67507), IBM Research Division, T. J. Watson Research Center, 1989 (unpublished).
6. J. Cullum, W. Kerner, and R. Willoughby, *Comput. Phys. Commun.* **53**, 19 (1989).
7. T. Ericsson and A. Ruhe, *Math. Comput.* **35**, 1251 (1980).
8. R. Freund, M. Gutknecht, and N. Nachtigal, Technical Report 90.45, RIACS, NASA Ames Research Center, 1990; *SIAM J. Sci. Stat. Comput.*, to appear.
9. B. Garbow, J. Boyle, J. Dongarra, and C. Moler (Eds). *Matrix Eigensystem Routines*, Lecture Notes in Computer Science, Vols. 6, 51 (Springer-Verlag, New York, 1976, 1977).
10. G. Golub and J. Ortega, *Scientific Computing and Differential Equations* (Academic Press, New York, 1991).
11. C. Lanczos, *J. Res. Nat. Bur. Stand.* **45**, 255 (1950).
12. M. Malik, *J. Comput. Phys.* **86**, 376 (1990).
13. M. Malik, private communication, 1991.
14. C. Moler and G. Stewart, *SIAM J. Numer. Anal.* **10**, 241 (1973).
15. N. Nayar, Ph.D. thesis, Applied Math., University of Virginia, 1992 (unpublished).
16. J. Ortega, *Matrix Theory: A Second Course* (Plenum, New York, 1987).
17. C. Paige, *J. Inst. Math. Appl.* **10**, 373 (1972).
18. B. Parlett, D. Taylor, and Z. Lin, *Math. Comput.* **44**, 104 (1985).
19. Y. Saad, *Linear Algebra Appl.* **34**, 269 (1980).
20. J. Wilkinson, *The Algebraic Eigenvalue Problem* (Oxford Univ. Press, London/New York, 1965).