EIGENPROBLEMS FOR MATRICES ASSOCIATED WITH PERIODIC BOUNDARY CONDITIONS

by

A. Bjorck
G. H. Golub

STAN-CS-75-486
MARCH 1975

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY

LINKÖPING UNIVERSITY
Department of Mathematics

Revised 1975-01-20

EIGENPROBLEMS FOR MATRICES ASSOCIATED WITH PERIODIC BOUNDARY CONDITIONS

Åke Björck and Gene H. Golub*

REPORT
LiH - MAT - R - **1974** - **8**

Issued jointly as Stanford University, Computer Science Report.

* The work of G. H. Golub was supported in part by National Science Foundation, GJ35135X and Atomic Energy Commission, AT(04-3)-326PA #30.

Professor Åke Björck
Department of Mathematics
Linköping University
S-581 83 Linköping, Sweden

Professor Gene H Golub
Computer Science Department
Stanford University
Stanford, California 94305, USA

Abstract

A survey of algorithms for solving the eigenproblem for a class of matrices of nearly tridiagonal form is given. These matrices arise from eigenvalue problem for differential equations where the solution is subject to periodic boundary conditions. Algorithms both for computing selected eigenvalues and eigenvectors and for solving the complete eigenvalue problem are discussed.

Key words: Eigenvalues, Periodic matrices.

AMS/MOS Classification: 65F05/65F15/65L15/65N25

1. Introduction

Eigenvalue problems of the form

$$\frac{d}{dx}\left(p(x)\frac{dy}{dx}\right)+ q(x)y + \lambda r(x)y = 0, \; p(x) > 0, \qquad (1.1)$$

with periodic coefficients

$$p(x+a) = p(x), \; q(x+a) = q(x), \; r(x+a) = r(x),$$

and where $y(x)$ is subject to periodic boundary conditions, arise in many practical application (see e.g. [10]). Using a second order difference approximation to (1.1), we are led to a matrix eigenvalue problem (see Evans [1])

$$Ax = xx, \qquad (1.2)$$

where A is a real, symmetric matrix of nearly tridiagonal form

$$A = \begin{pmatrix} a_1 & b_1 & & & b_n \\ b_1 & a_2 & b_2 & & 0 \\ & b_2 & \ddots & \ddots & \\ & 0 & \ddots & \ddots & b_{n-1} \\ b_n & & & b_{n-1} & a_n \end{pmatrix} \qquad (1.3)$$

In this paper we will give a survey of algorithms for solving a linear system of equations $Ax = c$, for computing selected eigenvalues and eigenvectors of A and for solving the complete eigenproblem (1.2). The algorithms can in general also be applied when A is a Hermitian matrix of the same structure as A in (1.3). Some of the methods presented are believed to be new.

We remark that differential equations in two space dimensions with periodic boundary conditions give rise to similar eigenvalue problems, where the matrix A now is nearly bloak-tridiagonal, i.e. a_i and b_i in (1.3) are replaced by square matrices A_i and B_i (see [3]). Many of the methods proposed will be relevant also to this more general problem.

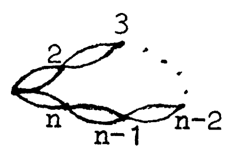
2. Basic transformations

We first remark that we can always assume in the following that

$$b_k \neq 0, \ k = 1,2\ldots,n. \tag{2.1}$$

Otherwise if $b_k = 0$, then, by a cyclic permutation of rows and columns these two elements can be brought to positions $(1,n)$ and $(n,1)$, and the matrix degenerates into a tridiagonal matrix.

The matrix (1.3) has no useful bandstructure at all. We note that the graph associated with A is a polygon:



From this graph it is quite easy to see, Martin[4], that the minimum bandwidth which can be obtained by permuting rows and columns of A is realized by ordering the nodes (n even)

$$1, \ n, \ 2, \ n-1, \ 3, \ n-2, \ldots, \frac{n}{2}+2, \ \frac{n}{2}, \ \frac{n}{1}+1.$$

(For n odd, the last three nodes are $(n-1)/2$, $(n+3)/2$, $(n+1)/2$.)
The permuted matrix then has the form (after renumbering the elements)

$$\tilde{A} = PAP^T = \begin{pmatrix} a_1 & b_1 & b_2 & & & & 0 \\ b_1 & a_2 & 0 & b_3 & & & \\ b_2 & 0 & a_3 & 0 & b_4 & & \\ & b_3 & \ddots & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 0 & & b_{n-1} \\ & 0 & & \ddots & 0 & a_{n-1} & b_n \\ & & & & b_{n-1} & b_n & a_n \end{pmatrix} \tag{2.2}$$

i.e. $\tilde{A}$ is symmetric and five-diagonal, with two inner diagonals almost zero. Rutishauser [5] has shown that A can be transformed further into tridiagonal form, using orthogonal similarity transformations. This can be accomplished with approximately $n^2/4$ plane rotations which corresponds to $\approx 6n^2$ multiplications (see Wilkinson [7] pp. 567-8). Unfortunately it is not possible to take advantage of the zeroes within the band of A, since these will rapidly fill up during the initial transformations.

Another useful observation is that A can be written as a rank one pertubation of a symmetric tridiagonal matrix

$$A = T + \sigma uu^T, \sigma = \pm b_n , \qquad (2.3)$$

where

$$T = \begin{pmatrix} a_1 \mp b_n & b_1 & & & \\ b_1 & a_2 & b_2 & & \\ & b_2 & \ddots & \ddots & \\ & & \ddots & \ddots & b_{n-1} \\ & & & b_{n-1} & a_n \mp b_n \end{pmatrix}, \quad u = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ \pm 1 \end{pmatrix} \qquad (2.4)$$

(Note that the first and last diagonal elements have been modified.) This splitting of A enables us to use some of the methods given by Golub [2].

Using perturbation theory the eigenvalues of A can be related to those of T in (2.4). If we denote these eigenvalues λ_k and d_k, k = 1,2...,n in decreasing order, then they satisfy the relations (Wilkinson [7] p. 98)

$$\lambda_k - d_k = 2\sigma m_k , \qquad (2.5)$$

where

$$0 \le m_k \le 1 , \sum_{k=1}^{m} m_k = 1 . \qquad (2.6)$$

Thus the eigenvalues λ_k separate the d_k at least in the weak sense, and if $\sigma \le 0$ then

$$d_1 \ge \lambda_1 \ge d_2 \ge \lambda_2 \cdots\cdot \ge d_n \ge \lambda_n .$$

Now from (2.1) it follows that the eigenvalues d_k are simple, and thus, the eigenvalues λ_k have at most multiplicity 2. Also if an eigenvalue λ_k has multiplicity 2, then d_k is also an eigenvalue of T. This differs from the tridiagonal case where if λ_k has multiplicity 2 at least one $b_i = 0$.

We remark that a simple example of a matrix A with eigenvalues of multiplicity 2 is the matrix $A = A(a,b)$ where

$$a_i = a,\ b_i = b,\ i = 1,2,\dots,n.$$

This matrix is a special case of a circulant and has the eigenvalues

$$\lambda_k = a + 2b\cos(2\pi k/n),\ k = 0,1,\dots,n-1.$$

All the eigenvalues of this matrix has multiplicity 2 except the eigenvalue $(a+2b)$ (and if n is even $(a-2b)$).

The special unsymmetric matrix

$$A = \begin{pmatrix} a_1 & b_1 & & & c_n \\ c_1 & a_2 & b_2 & & 0 \\ & c_2 & \ddots & \ddots & \\ & 0 & \ddots & \ddots & b_{n-1} \\ b_n & & & c_{n-1} & a_n \end{pmatrix},\quad b_i c_i > 0, \tag{2.6}$$

can often be reduced by a diagonal similarity to a symmetric matrix. If we take

$$\tilde{A} = D A D^{-1},\ D = \mathrm{diag}(d_i),$$

then $\tilde{A}$ is symmetric if

$$(d_k/d_{k+1})b_k = (d_{k+1}/d_k)c_k,\ k = 1,2,\dots,n.$$

where we have put d_{n+1}. Multiplying these relations together we find

$$\prod_{k=1}^{n} b_k = \prod_{k=1}^{n} c_k$$

If this relation is satisfied then d_k are determined by

$$d_1 = 1,\ d_{k+1} = \pm d_k(b_k/c_k)^{1/2},\ k = 1,\dots,n-1. \tag{2.7}$$

3. Linear system of equations

We first compare some different methods for solving $Ax = c$ when A is positive-definite.

3a) Gaussian elimination

When A is positive definite Gaussian elimination can be performed without pivoting. Hence symmetry is preserved with consequent savings in storage and operations. To avoid square roots we write the resulting factorization $A = LDL^T$ where

$$L = \begin{pmatrix} 1 & & & & \\ \gamma_1 & 1 & & & \\ & \gamma_2 & \ddots & & \\ & & \ddots & 1 & \\ \delta_1 & \delta_2 \cdots & \delta_{n-1} & & 1 \end{pmatrix}, \quad D = \mathrm{diag}(\alpha_i).$$

The elements in L and D are computed by the recursion formulas

$$\alpha_1 = a_1,\ \gamma_{k-1} = b_{k-1}/\alpha_{k-1},\ \alpha_k = a_k - \gamma_{k-1}b_{k-1}$$
$$k = 2,\ldots,n-1$$
$$\beta_1 = b_n,\ \beta_k = -\gamma_{k-1}\beta_{k-1},\ k = 2,\ldots,n-2 \qquad (3.1)$$
$$\beta_{n-1} = b_{n-1} - \gamma_{n-2}\beta_{n-2}.$$
$$\alpha_n^{(1)}c_1 = a_n,\ \delta_{k-1} = \beta_{k-1}/\alpha_{k-1},\ \alpha_n^{(k)} = \alpha_n^{(k-1)} - \delta_{k-1}\beta_{k-1},$$
$$\alpha_n = \alpha_n^{(n)}, \qquad k = 2,\ldots,n.$$

Thus, the complete decomposition requires $3n$ multiplications and $2n$ divisions. To solve the system $Ax = c$, we then have to solve the two triangular systems

$$Ly = c,\ L^Tx = D^{-1}y,$$

which requires another $5n$ multiplicative operations. This algorithm uses a total of $10n$ operations and n extra storage locations.

Gaussian elimination can also be applied to the reordered matrix (2.2). It is easy to verify that this requires the same number of operations and amount of storage as the algorithm above.

3b) Odd-even reduction

Assume first that n is even, and rearrange the rows and columns of $A = A_1$ through a permutation matrix

$$P_1 A_1 P_1^T = \left(\begin{array}{cccc|cccc}
a_2 & & & & b_1 & b_2 & & \\
 & a_4 & & & & b_3 & b_4 & \\
 & & \ddots & & & & \ddots & \ddots \\
 & & & & & & & b_{n-2} \\
 & & & a_n & b_n & & & b_{n-1} \\
\hline
b_1 & & & b_n & a_1 & & & \\
b_2 & b_3 & & & & a_3 & & \\
 & b_4 & \ddots & & & & \ddots & \\
 & & b_{n-2} & b_{n-1} & & & & a_{n-1}
\end{array}\right)$$

We now eliminate all even variables from all odd numbered equations by Gaussian elimination (Note that $P_1A_1P_1^T$ also is positive-definite). If we introduce the notation $c = c_1$, $x = x_1$ and

$$P_1A_1P_1^T = \begin{pmatrix} D_1 & B_1^T \\ B_1 & E_1 \end{pmatrix}, \quad P_1c_1 = \begin{pmatrix} d_1 \\ e_1 \end{pmatrix}, \quad P_1x_1 = \begin{pmatrix} y_1 \\ x_2 \end{pmatrix}, \tag{3.2}$$

then we get for x_2 after this elimination step the reduced system of order n/2

$$A_2x_2 = c_2 \tag{3.3}$$

where

$$A_2 = E_1 - B_1D_1^{-1}B_1^T, \quad c_2 = e_1 - B_1D_1^{-1}d_1 . \tag{3.4}$$

If x_2 is known, then we get y_1 by backsubstitution

$$y_1 = D_1^{-1}d_1 - D_1^{-1}B_1^Tx_2 \tag{3.5}$$

We note that this elimination step can be performed without using extra storage, since we can let

$$B_1 D_1^{-1}\ , \quad A_2,\ D_1^{-1}\ d_1,\ c_2$$

overwrite

$$B_1,\ \ D_1 \text{ and } E_1,\ d_1,\ e_1,$$

Taking symmetry into account, this first step requires $\approx 5n/2$ operations for computing $B_1 D_1^{-1}$ and A_2 and a further 2n operations for $D_1^{-1}\ d, c_2$ and y_1.

The important thing to note is that A_2 is again a symmetric tridiagonal matrix, with elements added in the lower left and upper right hand corners. Thus, if n is a power of two, then we can use the same decomposition repeatedly. Then we will obtain x in

$$n(9/2 + 9/4 + 9/8 + ...) \approx 9n$$

operations.

The odd-even method can in fact be applied without restriction on n, since if n is odd, then we get after the first reordering

$$P_1 A_1 P_1^T = \left(\begin{array}{cccc|cccc}
a_2 & & & & b_1 & b_2 & & \\
 & a_4 & & & & b_3 & b_4 & \\
 & & \ddots & & & & \ddots & \ddots \\
 & & & a_{n-1} & & & b_{n-2} & b_{n-1} \\
\hline
b_1 & & & & a_1 & & & b_n \\
b_2 & b_3 & & & & a_3 & & \\
 & b_4 & \ddots & b_{n-2} & & & \ddots & \\
 & & & b_{n-1} & b_n & & & a_n
\end{array}\right)$$

and obviously the reduced matrix A_2 will be of the same form also in this case. Perhaps the main advantage with the odd-even reduction is that it does not require extra storage. The operation count is also slightly lower than for Gaussian elimination, but this might be upset by the need for more organisational instructions. It should be pointed out that if $a_k = a$ and $b_k = b$ for all k there is a considerable simplification in the algorithm.

3c) Rank-one modification

By (2.3) and the Sherman-Morrison formula the solution to $Ax = c$ can be written as a linear combination of the solution to two tridiagonal systems of equations,

$$x = T^{-1}c - \beta T^{-1}u, \; \beta = \sigma u^T T^{-1} c/(1+\sigma u^T T^{-1} u). \qquad (3.6)$$

Since T is symmetric we have

$$u^T(T^{-1}c) = (T^{-1}u)^T c \; ,$$

and therefore we can also write (3.6) as

$$T\text{-}x = c - \beta u, \; \beta = v^T c/(1 + v_1 + v_n), \qquad (3.7)$$

where we solve for v from

$$Tv = \sigma u. \qquad (3.8)$$

(Note that in (3.7) the right hand side of the system of equations is modified only in the first and last component).
From the relations (2.5) and (2.6) it follows that if $\sigma \leq 0$, then

$$\lambda_{min}(T) \geq \lambda_{min}(A),$$

When T is positive definite the two systems of equations in (3.7) and (3.8) can be solved without pivoting, and we can compute c with a total of $9n$ multiplications . As in Gaussian elimination we need one extra temporary storage vector.

An important case when the algorithms given above are not directly applicable is in inverse iteration for computing eigenvectors of A. Then we want typically, to solve, the system of equations

$$(A - \lambda I)x_{k+1} = x_k \; ,$$

where A approximates an eigenvalue of A. Here the matrix (A - AI) is not in general positive-definite, and in Gaussian elimination pivoting must be used to preserve stability. Symmetry will then be destroyed, and therefore the algorithms given below for this case also apply when A is unsymmetric.

3d) Gaussian elimination with pivoting

This algorithm has been described in detail by Evans [1]. Here we prefer a slightly different approach. In [8 3, contribution I/6, it has been described how partial pivoting can be performed so that we get as a by-product the leading principal minors of A. This can be an advantage if selected eigenvalues of A are to be determined, and involves no more arithmetic than the more usual algorithm. The first (n-l) elimination steps will then require 3n multiplications, and the last between 3n and 4n multiplications (depending on the number of interchanges). The forward- and backsubstitution part will take between 5n and 6n, giving a total of less than 14n multiplications.

We note that Gaussian elimination with pivoting can also be applied to the five-diagonal matrix in (2.2). If we don't try to keep track of the zeroes within the band, then a standard procedure for general band-matrices (I/6 in [8]) can be used. This will however require 17n multiplications for the solution, and also more indexing operations.

3e) Rank-one modification, unsymmetric case

The formulas (3.7) and (3.8) apply also in this case, but we must now use pivoting when solving for v and x. This will increase the operation count to 12n multiplications. We point out, that this method cannot now be expected always to work. Irrespective of the sign of σ, T can now be much worse conditioned than A. We return to this question when discussing inverse iteration in section 5.

4. The complete eigenvalue problem

We now consider algorithms for computing the complete set of eigenvalues and possibly also the corresponding eigenvectors. We first note that unfortunately-the QR-algorithm cannot efficiently be applied directly to A. If we determine an orthogonal transformation Q^T such that $Q^TA = R$ is upper triangular, then R has the form:

$$\begin{pmatrix} x & x & & & x \\ & x & x & & x \\ & & x & x & x \\ & & & x & x \\ & & & & x \end{pmatrix} \qquad (n = 5)$$

It is easily seen that $A' = RQ = RAR^{-1}$ is a full matrix.

4a) Reduction to tridiagonal form

One approach is to reduce A by permutations and plane rotations to tridiagonal form as desribed in section 2. Then the efficient QR-algorithm (see [8] contribution II/3 and II/4) can be applied. The amount of work in the reduction ($6n^2$ multiplications) is less than that required by the &R-algorithm, which is $\approx 12n^2$ if only eigenvalues are computed and $\approx 4n^3$ if also eigenvectors are needed.

4b) Rank-one modification

If we assume that the eigenvalue problem for the tridiagonal matrix T in (2.3) has been solved, then

$$T = A - \sigma uu^T = Q\,D\,Q^T,\ D = \text{diag}(d_i) \qquad (4.1)$$

and thus

$$Q^TA\,Q = D + \sigma vv^T,\ v = Q^Tu. \qquad (4.2)$$

We now have to solve the eigenproblem for a diagonal matrix modified by a matrix of rank one, This problem has been discussed in [2]. We have for $\lambda \neq d_i$, $i = 1,2,\ldots,n$

$$\det(D + \sigma vv^T - AI) = \det(D - \lambda I)(1 + \sigma v^T(D - \lambda I)^{-1}v)$$

Thus, the characteristic polynomial is

$$p_n(\lambda) = \prod_{i=1}^{n}(d_i - \lambda) + \sigma\sum_{i=1}^{n} v_i^2 \prod_{\substack{j=1 \\ j\neq i}}^{n}(d_j - \lambda) \tag{4.3}$$

if $\lambda \neq d_i$, $i = 1,2,\ldots,n$, and

$$p_n(d_k) = \sigma\, v_k^2 \prod_{j=1}^{n}(d_j - d_k), \quad k = 1,2,\ldots,n. \tag{4.4}$$

Since from the assumption (2.1) it follows that the eigenvalues d_i of T are distinct, (4.4) implies that d_k is an eigenvalue of A if and only if $v_k = 0$. The corresponding eigenvector of A is then

$$x_k = Qe_k = q_k ,$$

i.e. it equals the eigenvector of T. In practical computation if we find that $v_k = \varepsilon$, then with $\lambda = d_k$ and $x_k = Qe_k$ we have

$$||Ax_k - \lambda x_k||_2 = ||(D + \sigma vv^T)e_k - d_k e_k||_2 = \sqrt{2}|\sigma\varepsilon| .$$

Thus, when ε is of the same order of magnitude as the uncertainties in the elements of A we can accept d_k and q_k as an eigenvector-eigenvalue pair of A.

The remaining eigenvalues of A can be computed by finding the roots of the equation

$$\omega(\lambda) = 1 + \sigma \sum_{i=1}^{n} v_i^2/(d_i-\lambda) = 0. \tag{4.5}$$

. The equation (4.5) can easily be solved, since we have precise bounds on each of the roots (from (2.5) and (2.6)). It is also easy to compute derivatives of w(h), so e.g. Newton's method may be used. When an eigenvalue λ_k of A is known, we have for the corresponding eigenvector the explicite expression

$$x_k = Q\,(D - \lambda_k I)^{-1}v. \tag{4.6}$$

The eigenvalues d_k of T can be efficiently computed by the QR-method. Note that when all the eigenvectors of A are not wanted, then it is not necessary to compute the whole matrix Q, but only the vector $v = Q^T u$.

The two methods described in this section requires about the same amount of work. However, an example where the last method is advantageous to use has been given in [10]. There the matrix T is real symmetric, but A has complex elements

$$A = T + \sigma u u^* , \quad u^* = (1,0,. . . . ,0,e^{i\phi}).$$

5. Computing selected eigenvalues and eigenvectors

If only a few eigenvalues and eigenvectors are required, then unless n is very small transformation to tridiagonal form or solution of the complete eigenproblem for T becomes to expensive. We consider here algorithms for computing selected eigenvalues based on the Sturm property of the sequence of leading minors $p_1(A)$, $p_2(\lambda),\ldots,\hat{p}_n(\lambda)$ of $(A - \lambda I)$. The corresponding eigenvectors can then be obtained by inverse iteration.

Following Evans [1] we define $p_i(\lambda)$ and $r_i(\lambda)$ as principal subdeterminants

$$p_i(\lambda) = \det\,(T[1,i]),\quad r_i(\lambda) = \det\,(T[n-i,n-1]) \qquad (5.1)$$

where T = T(A) is the tridiagonal matrix

$$T = \begin{pmatrix} a_1-\lambda & b_1 & & 0 \\ b_1 & a_2-\lambda & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ 0 & & b_{n-1} & a_n-\lambda \end{pmatrix}$$

Expanding the determinant $\hat{p}_n(\lambda) = \det\,(A-\lambda I)$ by the last column we get

$$\hat{p}_n(\lambda) = p_n(\lambda) - b_n^2\, r_{n-2}(\lambda) + 2(-1)^{n-1} b_1 b_2 \cdots b_n \qquad (5.2)$$

Then, the number of disagreements in sign between consecutive numbers in the sequence $p_0,\ldots,p_{n-1}(\lambda),\hat{p}_n(\lambda)$ is equal to the number of eigenvalues smaller than λ. To avoid difficulties with underflow and overflow one usually instead computes the ratios of succesive numbers in this sequence. If we divide (5.2) by $p_{n-1}(\lambda)$ then since $p_{n-1}(\lambda) = r_{n-1}(\lambda)$ we get

$$\hat{q}_n(\lambda) = q_n(\lambda) - b_n^2/s_{n-1}(\lambda) + 2b_n \cdot t_n \qquad (5.3)$$

where

$$q_i(\lambda) = p_i(\lambda)/p_{i-1}(\lambda)\ ,\quad s_i(\lambda) = r_i(\lambda)/r_{i-1}(\lambda),$$

$$t_n = p_{n-1}^{-1}(\lambda) \cdot \prod_{i=1}^{n-1}(-b_i) = \prod_{i=1}^{n-1}(-b_i/q_i(\lambda)) .$$

To compute (5.3) we use the recursions

$$\begin{aligned}
&q_1 = a_1-\lambda \ , \quad q_i = a_i-\lambda - b_{i-1}^2/q_{i-1} \ , \quad i=2,\ldots,n, \\
&s_1 = a_{n-1}-\lambda \ , \quad s_i = a_{n-i}-\lambda - b_{n-i}^2/s_{i-1}, \quad i=2,\ldots,n-1, \qquad (5.4) \\
&t_1 = b_n, \qquad t_i = -t_{i-1} \cdot b_{i-1}/q_{i-1}, \quad i=2,\ldots,n.
\end{aligned}$$

As in contribution II/5 in [8] we merely replace a zero $q_{i-1}(\lambda)$ or $s_{i-1}(\lambda)$ by a suitable small positive quantity. The number of negative elements in the sequence $q_1(\lambda),\ldots,q_{n-1}(\lambda), \hat{q}_n(\lambda)$ is now equal to the number of eigenvalues smaller than λ. The computation of this sequence using (5.3) and (5.4) takes 2n divisions and 2n multiplications if b_i^2 are computed once and for all. This is more than for the tridiagonal case but much less than for the similar algorithm by Evans [1].

The formulas (5.1) and (5.3) are not always suitable when det $(A-\lambda I)$ has a double zero λ^*. Then λ^* is also a simple zero of $s_{n-1}(\lambda)$ and $q_{n-1}(\lambda)$ and we will get cancellation of high order in (5.3). We now derive an algorithm which although not unconditionally stable, performs well also for double roots.

If we apply Gaussian elimination without pivoting to the matrix $(A-\lambda I)$, then before the $(i-1)$:st step we have the reduced matrix

$$\begin{pmatrix}
q_{i-1} & b_{i-1} & & t_{i-1} \\
b_{i-1} & (a_i-\lambda) & b_i & 0 \\
 & \ddots & \ddots & \\
0 & & \ddots & b_{n-1} \\
t_{i-1} & & b_{n-1} & c_{i-1}
\end{pmatrix}$$

Here for $i=1,2,\ldots,n-1$, q_i, t_i and c_i are determined by the recursion formulas (5.4) and

$$c_1 = a_n-\lambda, \quad c_i = c_{i-1} - t_{i-1}^2/q_{i-1}. \qquad (5.5)$$

After (n-2) elimination steps, we end up with the 2x2 matrix

$$\begin{pmatrix} q_{n-1} & b_{n-1}+s_{n-1} \\ b_{n-1}+s_{n-1} & c_{n-1} \end{pmatrix} \tag{5.6}$$

and thus

$$q_n(\lambda) = c_{n-1}(\lambda) - (b_{n-1} + s_{n-1}(\lambda))^2/q_{n-1}(\lambda) \quad . \tag{5.7}$$

Here, in case λ^* is a double zero, all elements in the matrix (5.6) also equals zero for $\lambda=\lambda^*$, and thus no cancellation occurs in (5.7). Unfortunately (5.5) is not a stable way of computing c_{n-1}.

When $q_{i-1}(\lambda)$ is small, then $|c_i(\lambda)| >> |c_{i-1}(\lambda)|$ and severe cancellation which causes instability will take place in the later steps. However, unless $b_{i-1}^2/(a_i-\lambda)$ also is small, we have $|c_{i+1}| << |c_i|$ and we can avoid the cancellation by taking a double step

$$c_{i+1} = c_{i-1} - t_{i-1}^2(a_i-\lambda)/(q_i \cdot q_{i-1}) \tag{5.8}$$

whenever $|q_i(\lambda)| >> |a_i-\lambda|$. Similarly if $q_{n-2}(\lambda)$ is close to zero we have to modify (5.7). By combining the last two steps we get

$$\hat{q}_n = c_n - b_n^2/q_{n-1} + 2\, t_{n-2}\, b_{n-2}\, b_{n-1}/(q_{n-1}q_{n-2}), \tag{5.9}$$

with c_n defined by (5.8). As before we can replace the zero $q_{i-1}(\lambda)$ by a small positive quantity $\varepsilon|b_{i-1}|$, where ε is the relative precision of the arithmetic which is used. The operation count for this algorithm is the same as for the first one, but the overhead is slightly larger.

Recursion formulas similar to these above can also be developed for the five-diagonal form (2.2). However, the formulas corresponding to (5.3) and (5.4) become more complicated, and they retain the same shortcomings.

Neither of the algorithm given above is without objections. To compute the ratios of successive minors of (A-λI) incompletely satisfactory way, it seems that we have to use the elimination algorithm mentioned under 3d. This algorithm requires however 7n multiplicative operations and thus is not quite as efficient as the other two. If also ... vectors are needed ...

algorithm can be used in the inverse iterations for the eigenvectors.

An alternative to the methods described so far, is reordering to five-diagonal form and using the &R-algorithm for band symmetric matrices (see[8] , contribution II/7). Note that this procedure is recommended only for computing selected eigenvalues and not for solving the complete eigenproblem.

We finally discuss methods based on the rank one modification (2.3) of A. We have

$$\det(A - \lambda I) = \det(T - \lambda I)\ \det(I + \sigma(T - \lambda I)^{-1}uu^T) =$$
$$\det(T - \lambda I)(1 + \sigma u^T(T - \lambda I)^{-1}u).$$

Here, we cannot as in section 4 exclude the case when λ is an eigenvalue of both A and T. Thus we cannot divide out $\det(T - \lambda I)$ and the characteristic equation becomes

$$\hat{p}_n(\lambda) = \det(T - \lambda I)(1 + \sigma u^T v(\lambda)) = p_n(\lambda)\ \omega(\lambda)\ , \qquad (5.10)$$

where $v(\lambda)$ is the solution to the tridiagonal system

$$(T - \lambda I)v(\lambda) = u\ . \qquad (5.11)$$

To solve (5.11) for $v(\lambda)$ and compute $\omega(\lambda)$ requires 8n operations, and since $p_n(\lambda)$ is the determinant of a tridiagonal matrix it can be computed from the usual recursion formula in 2n operations. Note that since

$$\omega'(\lambda) = \sigma v^T v\ ,$$

Newton's method can be applied with little extra work.

We now turn to the computation of selected eigenvectors, assuming that accurate approximations for the corresponding eigenvalues have been computed by one of the algorithms outlined above. This is usually best done by inverse iteration

$$(A - \lambda I)z_{r+1} = x_r, \quad x_{r+1} = z_{r+1}/\|z_{r+1}\|_2\ , \ r = 0,1,\ldots \qquad (5.12)$$

The choice of x_0 here requires some care. A very complete discussion of this choice and the other properties of this process has been given by Wilkinson[9] . To solve (5.12) we can use one of the methods for indefinite systems given in section 3. Gaussian elimination with pivoting is straighforward to use, and is recommended when the eigenvalues have been found by the &R-algorithm or the Sturm sequence methods.

When the eigenvalues have been found by solving (5.10), then the rank one modification technique can be used also in the inverse iterations, Using (3.6) the solution to (5.12) can be written

$$z_{r+1} = y_{r+1} - \beta v \ , \ \beta = \sigma u^T y_{r+1}/\omega(\lambda) \ ,$$

where $v = v(\lambda)$ is defined by **(5.11)** and y_{r+1} by

$$(T - \lambda I) y_{r+1} = x_r. \qquad (5.13)$$

Since $\omega(\lambda)$ may be close to zero, we get a more appropriate scaling by instead considering the vector

$$\hat{z}_{r+1} = \omega(\lambda)\, y_{r+1} - u^T y_{r+1}\, v \ . \qquad (5.13)$$

Now, assume that λ is a very good approximation to an eigenvalue of A, which is not an eigenvalue of T. Then, it follows that $p_n(\lambda) \neq 0$ and w(A) ≈ 0, and that v(X) will be a good approximation to the corresponding eigenvector. Thus, the eigenvector is obtained already from (5.11) when solving for the eigenvalue, and no inverse iteration has to be done.

In the case when λ is an eigenvalue of both A and T, then we have seen in section 3 that also the corresponding eigenvectors must coincide. Then we must have $u^T q = 0$, where q is this eigenvector, and A/q will -be an eigenvalue/eigenvector pair of the matrix $(T + \sigma u u^T)$ for arbitrary values of σ. It follows that in this case $\omega(\lambda)$ will remain bounded, but in general not equal to zero. We can obtain the eigenvector by applying inverse iteration to T, i.e. compute the sequence of vectors

$$y_{r+1} = (T - \lambda I)^{-1} x_r, \quad x_{r+1} = y_{r+1}/\|y_{r+1}\|_2 .$$

6. Conclusions

We have surveyed methods for solving the eigenvalue problem for nearly tridiagonal matrices of the form (1.3), which arise from periodic boundary problems. Although many of the standard methods can be made to work efficiently, it is surprising how much trouble the extra two non-zero elements generates. Two examples of this are that this matrix structure is not invariant under m-iterations, and that it requires much more work to generate the Sturm sequence than in the tridiagonal case. One should also point out that the simple backward analysis of rounding errors in the tridiagonal case does not generally carry over to matrices'of the form (1.3).

Acknowledgements

The work of the first author was done when he spent a sabbatical term at the Computer Science Department at Stanford University. He would like to thank all people at the department, Gene Golub especially, for making this stay such a pleasant one.

The first author also received financial support from the Wallenberg foundation. The second author was supported in part by an AEC and NSF grant.

References

1. D.J. Evans, Numerical solution of the Sturm-Liouville problem with periodic boundary conditions. Conference on Applications of Numerical Analysis. J.L. Morris (Ed.) Springer Verlag (1971).

2. G.H. Golub, Some modified matrix eigenvalue problems. SIAM Review, Vol. 15(1973), pp. 318-334.

3. G.H. Golub, L. Jennings and W.H. Yang, Waves in two dimensionally periodic structures. To appear in J. Computational Phys.

4. D.W. Martin, Linear Algebraic Equations in Numerical Analysis. An Introduction. J. Walsh (Ed.), Academic Press (1966).

5. H. Rutishauser, On Jacobi rotation patterns. Proc. of Symposia in Appl. Math. AMS Vol. 15 (1963), pp. 219-239.

6. G.W. Stewart, Modifying pivot elements in Gaussian Elimination. Math. of Comp. vol. 28(1974), pp. 537-542.

7. J.H. Wilkinson, The algebraic eigenvalue problem. Oxford University Press (1965).

8. J.H. Wilkinson and C. Reinsch, Handbook for Automatic Computation. Vol. II Springer Verlag (1971).

9. J.H. Wilkinson, Inverse iteration in theory and practice. Instituto Nazionale di Alta Matematica, Symposia Mathematica, Vol. X Bologna (1972).

10. W.H. Yang and E.H. Lee, Model analysis of Floquet waves in composite material. J. of Appl. Mechanics.