THE EIGENVALUES OF SYMMETRIC TRIDIAGONAL MATRIX METHOD WITH ACCELERATION _BY THE QR SHIFT

Dr. Wadia Faid Hassan Al-shameri
Thamar University, Faculty of Applied Science,
Department of Mathematics, Thamar Prov., Yemen.
E - Mail: wfha61@yahoo.com

Abstract :

This research investigates the application of the QR - method for computing all the eigenvalues of the real symmetric tridiagonal matrix. The Householder method will be used for reduction of the real symmetric matrix to symmetric tridiagonal form, and then the so-called QR - method with acceleration shift applies a sequence of orthogonal transformations to the symmetric tridiagonal matrix which converges to a similar matrix that is tridiagonal. This tridiagonal matrix possesses an eigenvalues similar to the eigenvalues of the symmetric tridiagonal matrix. Particular attention is paid to the shift technique that accelerates the rate of convergence. Computer algorithms for implementing the Householder's method and QR – method are presented. Computer Matlab programs for performing the Householder algorithm and the QR algorithm (with acceleration shift) are listed in the Appendix.

1. Introduction

In 1950s Alston Scott Householder[1] devised a method called Householder method [1] for reducing a given real symmetric n x n matrix A; that is, $A = A^t$ [2], to a similar symmetric matrix in tridiagonal form; that is, the only nonzero entries in the matrix lie either on the diagonal or on the subdiagonals directly above or below the diagonal. This method has a wide application in areas other than eigenvalues[3] approximation such as solving systems of linear equations. The method is much more stable than Gaussian elimination method. But this method takes more time to get the solution than Gaussian method does.

Section two of this research, introduces the Householder's method. This method is used to construct a real symmetric tridiagonal matrix B that is similar to the given real symmetric n x n matrix A.

It is well-known that (Theorem 9.10 of [1] and Theorem 8.39 of [2]) if A is an n x n symmetric matrix and D is diagonal matrix whose diagonal entries are the eigenvalues of A, then an orthogonal matrix4 Q exists with the property that D = Q-1AQ = QtAQ. A consequence of this is that the symmetric matrix A is similar to the diagonal matrix D. Because the matrix Q (and consequently D) is generally difficult to compute, Householder’s method offers a sequence of n-2 orthogonal transformations of the form PAP that will reduce A into a symmetric tridiagonal matrix. An orthogonal transformation of this form is called a Householder transformation (see [1], [2], [3], and [6]).

---

[1] 1904-1993.

[2] A^t means the transpose of the n x n matrix A.

[3] The n x n matrix A has precisely n, not necessarily distinct, eigenvalues (or characteristic value of the matrix A) that are the zeros of the characteristic polynomial $P(\lambda) = \det(A - \lambda I)$ of A.

[4] An n x n matrix Q is said to be an orthogonal matrix if $Q^{-1} = Q^t$ (i.e., $Q^t Q = I$, also $Q Q^t = I$).

The Householder's method can be translated into computer algorithm to find symmetric tridiagonal matrix as a Householder reduction of a real symmetric

n x n matrix A to symmetric tridiagonal form. An algorithm based on the Householder's method is known as the Householder algorithm. This algorithm will be presented here in section three, and can be found in [1]. Using the Householder algorithm we provide an example placed in section four which illustrates the procedure involved in the Householder algorithm.

The Householder's method is one of the methods based on similarity transformations. This method will be used to convert a symmetric matrix into a similar matrix that is tridiagonal. Methods such as QR – method[1] can then be applied to the tridiagonal matrix for finding all eigenvalues of a symmetric tridiagonal matrix (see [6, p 601]) to compute approximations to all eigenvalues.

Techniques such as the QR - method with acceleration shifts have been investigated in section five of this research. More generally, the QR - method is an efficient for the calculation of all eigenvalues of a square matrix whose entries are real numbers. If the matrix A is symmetric[2] tridiagonal matrix, then the QR - method can be used directly to find the eigenvalues of the symmetric tridiagonal matrix A. If the symmetric matrix A is not in tridiagonal form, the first step is to apply the Householder's method to compute a symmetric tridiagonal matrix similar to, and hence with the same eigenvalues as the given matrix A.

Although, the QR - method will produce the eigenvalues of symmetric tridiagonal matrix, but the rate of convergence is slow [6, p 603]. One easy way that speeds up the rate of convergence is to add

[1] The QR – method was introduced by Francis.

[2] If A is symmetric matrix, the eigenvalues of A are all real numbers [2, p.450]. This result will be considered here for approximation of the eigenvalues of the symmetric matrix A.

a shifting technique that will accelerate the rate of convergence when the QR – method with shifting [1, 6] is repeatedly iterated for producing all real eigenvalues of symmetric tridiagonal matrix.

There are several algorithms treating the QR – method [6], one of which we shall present in section six is named QR algorithm after the mathematician Francis, who introduced the QR – method (see [1]). The diagonal entries of the reduced matrix into symmetric tridiagonal form using the QR algorithm which is based on the QR – method with acceleration shifts are approximations to the eigenvalues of the given symmetric matrix.

The presentation of the QR algorithm should not be the end, an investigation which illustrates the steps of the QR algorithm has been chosen as an example calculation, contained in section seven, to compute all eigenvalues of a symmetric tridiagonal matrix that arises from the application of the QR algorithm.

A computer programs (or routines) were written in Matlab software perform the Householder algorithm and QR algorithm for reducing the real symmetric matrix into the symmetric tridiagonal form, and for computing all the eigenvalues of a symmetric tridiagonal matrix respectively. The main computational routine is found for implementing the Householder algorithm which calls the routine found for implementing the QR algorithm. The two programs listed in the Appendix, and those described and will also produce results cited in section eight. The results prove that the two computer algorithm programs work.

2. Householder's Method

Suppose that A is a real symmetric $n \times n$ matrix, Householder's method will be used to construct a similar symmetric tridiagonal matrix by n-2 orthogonal transformations. Each transformation annihilates the required part of a whole column and whole corresponding row. The basic ingredient is a Householder n×n matrix P which has the form

$P = I - 2ww^t$

where $w \in R^n$ is a real vector with $w^t w = 1$, is called the Householder transformation.

Now, we shall investigate the most important properties of the Householder transformation $P = I - 2ww^t$, $w \in R^n$.

The symmetry is from

$$P^t = (I - 2ww^t)^t = I^t - (2ww^t)$$
$$= I - 2(w^t)^t w^t = I - 2ww^t.$$

Further, the orthogonality is from

$$PP^t = (I - 2ww^t)(I - 2ww^t)$$
$$= I - 4ww^t + 4ww^t ww^t$$
$$= I - 4ww^t + 4w(w^t w)w^t$$
$$= I - 4ww^t + 4ww^t = I.$$

So, $P^{-1} = P^t = P$.

Consider again the real symmetric n×n matrix A, we shall follow [1], [3] and [6] to show that how one can construct a symmetric tridiagonal matrix $A^{(n-1)}$ similar to $A = A^{(1)}$ by applying the Householder method which determines a sequence of n-2 Householder transformations of the form PAP which will reduce $A = A^{(1)}$ to the symmetric tridiagonal matrix $A^{(n-1)}$.

The Householder method begins by determining a Householder transformation $P^{(1)}$ with the property that $A^{(2)} = P^{(1)} A P^{(1)}$ has $a_{j1}^{(2)} = 0$,

for each $j = 3,4, \dots ,n$, (2.1)

and by symmetry, $a_{1j}^{(2)} = 0$. The vector

$w = (w_1, w2,\dots, w_n)^t$ is chosen that $w^t w = 1$ so, equation (2.1) holds, and in the matrix $A^{(2)} =$

$P^{(1)} A P^{(1)} = (I- 2ww^t) A (I- 2ww^t)$, we have

$a_{11}^{(2)} = a_{11}$ and $a_{j1}^{(2)} = 0$, for each j = 3, 4, …, n. This choice imposes n conditions on the n unknown's w_1, $w_2, \dots, w_n$. Setting $w_1 = 0$ ensures that $a_{11}^{(2)} = a_{11}$. It is required that

$$P^{(1)} = I - 2ww^t$$

to satisfy

$$P^{(1)}(a_{11}, a_{21}, a_{31}, \dots, a_{n1})^t = (a_{11}, \alpha, 0, \dots, 0)^t \qquad (2.2)$$

where α will be chosen later. To simplify notation, let

$$\hat{w} = (w_2, w_3, \dots, w_n)^t \in R^{n-1},$$

$$\hat{y} = (a_{21}, a_{31}, \dots, a_{n1})^t \in R^{n-1}$$

and $\hat{P}$ be the (n-1) × (n-1) Householder transformation

$$\hat{P} = I_{n-1} - 2\hat{w}\hat{w}^t.$$

Equation (2.2) then becomes

$$P^{(n-1)}\begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ . \\ . \\ . \\ a_{n1} \end{bmatrix} = \begin{bmatrix} 1 & . & 0 & . & . & . & . & 0 \\ . & . & . & . & . & . & . & . \\ 0 & . & & & & & & \\ . & . & & & & & & \\ . & . & & & & & & \\ . & . & & & & \hat{P} & & \\ . & . & & & & & & \\ 0 & . & & & & & & \end{bmatrix}\begin{bmatrix} a_{11} \\ \dots\dots \\ \hat{y} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} \\ \dots\dots \\ \hat{P}\hat{y} \end{bmatrix} = \begin{bmatrix} a_{11} \\ \dots\dots \\ \alpha \\ 0 \\ . \\ . \\ . \\ 0 \end{bmatrix}$$

with

$$\hat{P}\hat{y} = (I_{n-1} - 2\hat{w}\hat{w}^t)\hat{y}$$

$$= \hat{y} - 2(\hat{w}^t\hat{y})\hat{w}$$

$$= (\alpha, 0, \dots, 0)^t. \qquad (2.3)$$

Let $r = \hat{w}^t\hat{y}$. Then

$$(\alpha, 0, \dots, 0)^t =$$

$(a_{21} - 2rw_2, a_{31} - 2rw_3, \ldots, a_{n1} - 2rw_n)^t$,

where w_i can be determined when we know α and r. Equating components gives

$\alpha = a_{21} - 2rw_2$ and $0 = a_{j1} - 2rw_j$, for each

$j = 3, \ldots, n$.

Thus,

$$2rw_2 = a_{21} - \alpha \qquad (2.4)$$

and

$$2rw_j = a_{j1}, \text{ for each } j = 3, \ldots, n. \qquad (2.5)$$

Squaring both sides of each of the equations (2.4) and (2.5) and adding gives

$$4r^2 \sum_{j=2}^{n} w_j^2 = (a_{21} - \alpha)^2 + \sum_{j=3}^{n} a_{j1}^2 .$$

Since $w^t w = 1$ and $w_1 = 0$, we have $\sum_{j=2}^{n} w_j^2 = 1$, and

$$4r^2 = \sum_{j=2}^{n} a_{j1}^2 - 2\alpha a_{21} + \alpha^2 . \qquad (2.6)$$

Using equation (2.3) and the fact that P is orthogonal imply that

$\alpha^2 = (\alpha, 0, \ldots, 0)(\alpha, 0, \ldots, 0)^t =$

$(\hat{P}\hat{y})^t \hat{P}\hat{y} = \hat{y}^t \hat{P}^t \hat{P}\hat{y} = \hat{y}^t \hat{y}$. Thus,

$$\alpha^2 = \sum_{j=2}^{n} a_{j1}^2$$

which, when substituted into equation (2.6), gives

$$2r^2 = \sum_{j=2}^{n} a_{j1}^2 - \alpha a_{21}.$$

To ensure that $2r^2 = 0$ only if

$$a_{21} = a_{31} = \ldots = a_{n1} = 0,$$

we choose

$$\alpha = -\operatorname{sgn}(a_{21})\left(\sum_{j=2}^{n} a_{j1}^2\right)^{1/2},$$

this implies that

$$2r^2 = \sum_{j=2}^{n} a_{j1}{}^2 + |a_{21}|\,(\sum_{j=2}^{n} a_{j1}^2)^{1/2}.$$

Using this choice of α and $2r^2$, we solve equations (2.2) and (2.6) to obtain

$$w_2 = \frac{a_{21} - \alpha}{2r} \quad \text{and} \quad w_j = \frac{a_{j1}}{2r} \;,$$

for each j = 3, … , n.

The choice of $P^{(1)}$ can be summarized as follows.

$$\alpha = -\text{sgn}(a_{21})(\sum_{j=2}^{n} a_{j1}^2)^{1/2}$$

$$r = (\frac{1}{2}\alpha^2 - \frac{1}{2}a_{21}\alpha)^{1/2},$$

$$w_1 = 0,\; w_2 = \frac{a_{21} - \alpha}{2r}, \quad \text{and} \quad w_j = \frac{a_{j1}}{2r}, \quad \text{for each } j = 3, ..., n.$$ This choice is used to obtain P, and hence $A^{(2)}$. Thus, the first Householder transformation is applied to the matrix $A^{(1)}$ = A and is denoted by

$$A^{(2)} = P^{(1)} A P^{(1)}$$

$$= \begin{bmatrix} a_{11}^{(2)} & a_{12}^{(2)} & 0 & \cdots & 0 \\ a_{21}^{(2)} & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(2)} & a_{n3}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix}.$$

The second Householder transformation is applied to the matrix $A^{(2)}$ and is denoted by $A^{(3)} = P^{(2)} A^{(2)} A^{(2)}$, the process is repeated for k = 3, …, n-2 as follows.

$$\alpha = -\text{sgn}(a_{k+1,k}^{(k)})(\sum_{j=k+1}^{n} (a_{jk}^{(k)})^2)^{1/2},$$

$$r = \left(\frac{1}{2}\alpha^2 - \frac{1}{2}\alpha a^{(k)}_{k+1,k}\right)^{(1/2)},\ w_1^{(k)} = w_2^{(k)} = \ldots = w_k^{(k)} = 0,\ w_{k+1}^{(k)} = \frac{a_{k+1,k}^{(k)} - \alpha}{2r},$$

$$w_j^k = \frac{a_{j,k}^k}{2r},\ \text{for each } j = k+2, k+3, \ldots, n,$$

$P^{(k)} = I - 2w^{(k)}(w^{(k)})^t$, and

$A^{(k+1)} = P^{(k)} A^{(k)} P^{(k)}$, where

$$A^{(k+1)} = \begin{bmatrix} a_{11}^{(k+1)} & a_{12}^{(k+1)} & 0 & \cdots & \cdots & \cdots & 0 \\ a_{21}^{(k+1)} & \ddots & \ddots & \ddots & & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 & \cdots & 0 \\ \vdots & \ddots & a_{k+1,k}^{(k+1)} & a_{k+1,k+1}^{(k+1)} & a_{k+1,k+2}^{(k+1)} & \cdots & a_{k+1,n}^{(k+1)} \\ \vdots & & 0 & \vdots & \ddots & & \vdots \\ \vdots & & \vdots & \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & a_{n,k+1}^{(k+1)} & \cdots & \cdots & a_{nn}^{(k+1)} \end{bmatrix}$$

Continuing in this manner, the symmetric tridiagonal matrix $A^{(n-1)}$ is constructed, where

$A^{(n-1)} = P^{(n-2)} P^{(n-3)} \ldots P^{(1)} A P^{(1)} \ldots P^{(n-3)} P^{(n-2)}$.

3. Householder Algorithm

Given a real n x n symmetric matrix A; to obtain a symmetric tridiagonal matrix $A^{(n-1)}$ similar to $A = A^{(1)}$, is to construct the n-2 Householder transformation $A^{(2)}, A^{(3)}, A^{(4)}, \ldots, A^{(n-1)}$, where $A^{(k)} = (a_{ij}^{(k)})$ for each k = 1, 2, 3, …, n-1, using the Householder’s method. The following algorithm known as Householder algorithm (see [1] for more details) performs the Householder's method presented in section two.

The Pseudo-code of the Householder algorithm is outlined as follows.

- Set dimension(A) = n;
- Set matrix A

% Construct the matrices $A^{(2)}, A^{(3)}, A^{(4)}, \ldots,$
$A^{(n-1)}$

- for k = 1 to n-2
- $q = \sum_{j=k+1}^{n} \left(a_{jk}^{(k)}\right)^2$
- if $a_{k+1,k}^{(k)} = 0$ then $\alpha = -q^{(1/2)}$

  else $\alpha = -\dfrac{q^{1/2} a_{k+1,k}^{(k)}}{\left|a_{k+1,k}^{(k)}\right|}$

% let mrs = $2r^2$

- $mrs = \alpha^2 - \alpha a_{k+1,k}^{(k)}$

% there is no need to $\upsilon_1 = \ldots = \upsilon_{k-1} = 0$

- $\upsilon_k = 0$

  $\upsilon_{k+1} = a_{k+1,k}^{(k)} - \alpha$

  for j = k+2 to n

  $\upsilon_j = a_{jk}^{(k)}$

  % w $= (\dfrac{1}{\sqrt{2mrs}}) v = \dfrac{1}{2r} v$.

  end (for loop)

- for j = k to n

  $u_j = \left(\dfrac{1}{mrs}\right) \sum_{i=k+1}^{n} a_{ji}^{(k)} v_i$

% $(\dfrac{1}{mrs}) A^{(k)} v = \dfrac{1}{2r^2} A^{(k)} v = \dfrac{1}{r} A^{(k)} w$.

- $mult = \sum_{i=k+1}^{n} \upsilon_i u_i$

% $mult = v^t u = u = \dfrac{1}{2r^2} v^t A^{(k)} v$

  end (for loop)

- for j = k to n
- $z_j = u_j - \left(\dfrac{M}{2mrs}\right) \upsilon_j$

% $z = u - \frac{1}{2mrs} v^t u v = u - \frac{1}{4r^2} v^t u v$

% $= u - ww^t u = \frac{1}{r} A^{(k)} w - ww^t \frac{1}{r} A^{(k)} w$

% construct $A^{(k+1)} = A^{(k)} - vz^t - zv^t = (I-2ww^t)$
$A^{(k)} (I - zww^t)$

- for l = k+1 to n-1
- for j = l+1 to n

$$a_{ji}^{(k+1)} = a_{ji}^{(k)} - \upsilon_i z_j - \upsilon_j z_i$$

$$a_{ij}^{(k+1)} = a_{ji}^{(k+1)}$$

- $a_{ii}^{(k+1)} = a_{ii}^{(k)} - 2\upsilon_i z_i$

end (for loop)

end (for loop)

- $a_{nn}^{(k+1)} = a_{nn}^{(k)} - 2\upsilon_n z_n$
- for j = k+2 to n

$$a_{kj}^{(k+1)} = a_{jk}^{(k+1)} = 0$$

end (for loop)

- $a_{k+1,k}^{(k+1)} = a_{k+1,k}^{(k)} - \upsilon_{k+1} z_k$

$$a_{k,k+1}^{(k+1)} = a_{k+1,k}^{(k+1)}$$

end (for loop)

- end ($A^{(n-1)}$ is symmetric, tridiagonal, and similar to $A = A^{(1)}$).

4. Illustrative Example

The 4×4 matrix

$$A = \begin{bmatrix} 4 & 1 & -2 & 2 \\ 1 & 2 & 0 & 1 \\ -2 & 0 & 3 & -2 \\ 2 & 1 & -2 & -1 \end{bmatrix}$$

is symmetric. To illustrate the procedure involved in the Householder algorithm to transform (or reduce) this matrix into a matrix that is symmetric and tridiagonal similar to A = $A^{(1)}$; we perform the following computations.

Set k = 1.

$$\alpha = -\mathrm{sgn}(a_{21})\left(\sum_{j=2}^{n} a_{j1}^2\right)^{1/2} = (-)(1+4+4)^{0.5} = -3;$$

$$r = \left(\frac{1}{2}\alpha^2 - \frac{1}{2}a_{21}\alpha\right)^{1/2} = \left(\frac{1}{2}(9) - \frac{1}{2}(1)(-3)\right)^{1/2} = \left(\frac{12}{2}\right)^{1/2} = \sqrt{6}\,;$$

$$w_1^{(1)} = 0\,;\ w^{(1)}{}_2 = \frac{a_{21}-\alpha}{2r} = \frac{1+3}{2\sqrt{6}} = \frac{\sqrt{6}}{3}\quad w_j^{(1)} = \frac{a_{j1}}{2r}\quad \text{for each j = 3,…., n;}$$

$$w_3^{(1)} = \frac{-2}{2\sqrt{6}} = \frac{-\sqrt{6}}{6}\,;\ w_4^{(1)} = \frac{2}{2\sqrt{6}} = \frac{\sqrt{6}}{6}\,;\ w^{(1)} = \left(0\ \frac{\sqrt{6}}{3}\ \frac{-\sqrt{6}}{6}\ \frac{\sqrt{6}}{6}\right)^t;$$

$$P^{(1)} = I - 2w^{(1)}w^{(1)t}$$

$$= I - 2\begin{pmatrix} 0 \\ \frac{\sqrt{6}}{3} \\ \frac{-\sqrt{6}}{6} \\ \frac{\sqrt{6}}{6} \end{pmatrix}\begin{pmatrix} 0 & \frac{\sqrt{6}}{3} & \frac{-\sqrt{6}}{6} & \frac{\sqrt{6}}{6} \end{pmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{-1}{3} & \frac{2}{3} & \frac{-2}{3} \\ 0 & \frac{2}{3} & \frac{2}{3} & \frac{1}{3} \\ 0 & \frac{-2}{3} & \frac{1}{3} & \frac{2}{3} \end{bmatrix};$$

$$A^{(2)} = P^{(1)}A^{(1)}\quad P^{(1)}$$

$$
= \begin{bmatrix} 4 & -2 & 0 & 0 \\ -3 & \frac{10}{3} & 1 & \frac{4}{3} \\ 0 & 1 & \frac{5}{3} & \frac{-4}{3} \\ 0 & \frac{4}{3} & \frac{-4}{3} & -1 \end{bmatrix}.
$$

Set k = 2.

$$
\alpha = -\operatorname{sgn}\left(a_{k+1,k}^{k}\right)\left(\sum_{j=k+1}^{n}\left(a_{jk}\right)^2\right)^{1/2} = -(1).((1)^2 + (4/3)^2)^{1/2} = -\frac{5}{3};
$$

$$
r = \left(\frac{1}{2}\alpha^2 - \frac{1}{2}\alpha a_{k+1,k}^{(k)}\right)^{1/2}
$$

$$
= \left(\frac{1}{2}(\frac{25}{9}) - \frac{1}{2}(\frac{-5}{3})(1)\right)^{1/2} = \frac{2\sqrt{5}}{3};
$$

$$
w_1^{(2)} = 0;\ w_2^{(2)} = 0;\ w_{k+1}^{(k)} = \frac{a_{k+1,k}^{(k)} - \alpha}{2r};\ w_3^{(2)} = \frac{1+5/3}{2.\frac{2\sqrt{5}}{3}} = \frac{2\sqrt{5}}{5};
$$

$w_j^{(k)} = \dfrac{a_{jk}^{(k)}}{2r}$ for each j = k+2, k+3, …, n; $w_4^{(2)} = \dfrac{4/3}{2.\frac{2\sqrt{5}}{3}} = \dfrac{\sqrt{5}}{5}$,

$$
w^{(2)} = (0 \quad 0 \quad \frac{2\sqrt{5}}{5} \quad \frac{\sqrt{5}}{5})^t.
$$

$$
P^{(2)} = I - 2\left(\frac{\sqrt{5}}{5}\right)^2 \begin{pmatrix} 0 \\ 0 \\ 2 \\ 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 2 & 1 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-3}{5} & \frac{-4}{5} \\ 0 & 0 & \frac{-4}{5} & \frac{3}{5} \end{bmatrix}.
$$

$$A^{(3)} = P^{(2)} A^{(2)} P^{(2)} = \begin{bmatrix} 4 & -3 & 0 & 0 \\ -3 & \frac{10}{3} & \frac{-5}{3} & 0 \\ 0 & \frac{-5}{3} & \frac{-33}{25} & \frac{68}{75} \\ 0 & 0 & \frac{68}{75} & \frac{149}{75} \end{bmatrix}.$$

The matrix $A^{(3)}$ is symmetric tridiagonal, and similar to the symmetric matrix $A = A^{(1)}$.

5. QR - method

Suppose that A is a real symmetric matrix. In the preceding sections we saw how Householder's method is used to construct a similar tridiagonal matrix. In this section QR – method will be used to find all eigenvalues of the symmetric tridiagonal matrix (see [6, p 601]). The QR - method has proved very efficient and robust and has practically outperformed all other methods [3]. If the original matrix is not symmetric, it is recommended first to transform it to Hessenberg matrix form[1] (In the symmetric case a tridiagonal matrix would be obtained.), then the QR - method will be applied for finding all eigenvalues of a general n x n real matrix, but it takes many iterations and becomes time consuming [7]. Plane rotations [2, pp.115-117] will be used to construct an orthogonal matrix $Q^{(i)}$ and an upper-triangular matrix $R^{(i)}$. The important step of the QR - method is QR factorization[2] $A^{(i)} = Q^{(i)} R^{(i)}$ and

---

[1] An n x n matrix A with $a_{i,j} = 0$ for $j > i + 1$ is called upper Hessenberg matrix; also when $a_{i,j} = 0$ for $i > j+1$, then the matrix A is called lower Hessenberg matrix.

[2] The QR factorization problem is defined as follows. Given $A \subseteq R^{mxn}$, $m >= n$, $rank(A) = n$, compute $Q \subseteq R^{mxm}$ and $R \subseteq R^{mxn}$ that satisfy $A = QR$, $Q^t Q = I$, $r_{i,j} = 0$, for all $i > j$ where $r_{i,j}$ is an individual

iteration $A^{(i+1)} = R^{(i)} Q^{(i)} = (Q^{(i)})^t A^{(i)} Q^{(i)}$, taken into account that the matrix $A^{(i+1)}$ is in tridiagonal form. The QR factorization arises in many applications like solving the least squares problem, eigenvalue decomposition, singular value decomposition etc.

We shall follow [1], [3], [4], [6], and [7] to investigate how the QR - method applies a sequence of orthogonal transformations $Q^{(i)}$ to the symmetric tridiagonal matrix obtained by the Householder's method.

Suppose that the symmetric tridiagonal matrix A is written as

$$A = \begin{bmatrix} a_1 & b_2 & 0 & \cdots & 0 \\ b_2 & a_2 & b_3 & \ddots & \vdots \\ 0 & b_3 & a_3 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & b_n \\ 0 & \cdots & 0 & b_n & a_n \end{bmatrix}. \qquad (5.1)$$

If $b_2 = 0$ or $b_n = 0$, then the 1×1 matrix $[a_1]$ or $[a_n]$ has an eigenvalues a_1 or a_n of A. If $b_j = 0$ for some j, where $2 < j < n$, then the problem can be reduced to considering, instead of A, the smaller matrices are

$$\begin{bmatrix} a_1 & b_2 & 0 & \cdots & 0 \\ b_2 & a_2 & b_3 & \ddots & \vdots \\ 0 & b_3 & a_3 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & b_{j-1} \\ 0 & \cdots & 0 & b_{j-1} & a_{j-1} \end{bmatrix}$$

and

$$\begin{bmatrix} a_j & b_{j+1} & 0 & \cdots & 0 \\ b_{j+1} & a_{j+1} & b_{j+2} & \ddots & \vdots \\ 0 & b_{j+2} & a_{j+2} & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & b_n \\ 0 & \cdots & 0 & b_n & a_n \end{bmatrix}. \qquad (5.2)$$

---

component of R.

Now, suppose that $b_j \neq 0$ for all j, the QR - method produces a sequence of matrices $A^{(1)} = A$, $A^{(2)}$, $A^{(3)}$, ... , as follows.

- $A^{(1)} = A$ is factored as a product $A^{(1)} = Q^{(1)} R^{(1)}$ where $Q^{(1)}$ is orthogonal matrix, and $R^{(1)}$ is upper triangular matrix.
- $A^{(2)}$ is defined as $A^{(2)} = Q^{(1)} R^{(1)}$.

In general, construct the orthogonal matrix $Q^{(i)}$ and upper-triangular matrix $R^{(i)}$ so that

$$A^{(i)} = Q^{(i)} R^{(i)}.$$

Then define

$$A^{(i+1)} = R^{(i)} Q^{(i)} .$$

Orthogonality of $Q^{(i)}$ implies that

$R^{(i)} = (Q^{(i)})^t A^{(i)}$, and

$$A^{(i+1)} = R^{(i)} Q^{(i)} = ((Q^{(i)})^t A^{(i)}) Q^{(i)}$$
$$= (Q^{(i)})^t A^{(i)} Q^{(i)}. \qquad (5.3)$$

This implies that $A^{(i+1)}$ and $A^{(i)}$ are similar. An important consequence is that $A^{(i)}$ is similar to A and hence has the same structure. Specifically, one can conclude that if A is tridiagonal then $A^{(i)}$ is also tridiagonal for all i. The process is continued, and by induction, $A^{(i+1)}$ has the same eigenvalues as the original matrix A, and $A^{(i+1)}$ becomes a diagonal matrix with the eigenvalues of A along its diagonal.

In order to construct the matrices $Q^{(i)}$ and $R^{(i)}$, a rotation matrix[1] will be used.

It is obvious that, for any rotation matrix P, the matrices AP and PA differs from A only in the i^{th} and j^{th} columns, and in the i^{th} and j^{th} rows respectively. For any $i \neq j$, the angle φ can be chosen so that the product

[1] A rotation matrix P differs from the identity matrix in at most four elements. These four elements are of the form $p_{ii} = p_{jj} = \cos\varphi$, and $p_{ij} = - p_{ji} = \sin\varphi$ for some φ and some $i \neq j$.

PA has a zero element for $(PA)_{ij}$. Since each plane rotation is presented by an orthogonal matrix, $PP^t = I$.

The factorization of $A^{(1)}$ into $A^{(1)} =$ $Q^{(1)} R^{(1)}$ uses a product of n-1 rotation matrices to construct

$$R^{(1)} = P_n P_{n-1} \dots P_2 \ A^{(1)}$$

The first step is to choose the rotation matrix P_2 with $p_{11} = p_{22} = \cos\varphi_2$ *and* $p_{12} = -p_{21} = \sin\varphi_2$, where

$$\sin\varphi_2 = \frac{b_2}{\sqrt{b_2^2 + a_1^2}}, \ \ and \ \ \cos\varphi_2 = \frac{a_1}{\sqrt{b_2^2 + a_1^2}}.$$

Then the matrix

$$A_2^{(1)} = P_2 \ A^{(1)}$$

has a zero in the (2, 1) subscript, since the (2, 1) subscript in $A_2^{(1)}$ is

$$(-\sin\varphi_2)a_1 + (\cos\varphi_2)b_2 = \frac{-b_2 a_1}{\sqrt{b_2^2 + a_1^2}} + \frac{a_1 b_2}{\sqrt{b_2^2 + a_1^2}} = 0.$$

Since the multiplication $P_2 \ A^{(1)}$ affects both rows 1 and 2 of $A^{(1)}$, the new matrix does not necessarily retain zero elements in the positions (1, 3), (1, 4),… and (1, n). However, $A^{(1)}$ is tridiagonal, so the (1, 4),…, (1, n) subscripts of $A_2^{(1)}$ must be 0. Only the (1, 3) subscript, the one in the first row and third column, can become nonzero.

In general, the plane rotation matrix P_k that reduces to zero the element of A in position (k, k-1); that is, $A_k^{(1)} = P_k \ A_{k-1}^{(1)}$. Continuing in similar way, a plane rotation matrix P_{k+1} that will reduce to zero the element of P_k located in position (k-1, k+1). The matrix $A_k^{(1)}$ has the form

$$A_k^{(1)} = \begin{bmatrix} z_1 & q_1 & r_1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & & & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & & & \vdots \\ \vdots & \ddots & 0 & z_{k-1} & q_{k-1} & r_{k-1} & \ddots & & \vdots \\ \vdots & & \ddots & 0 & x_k & y_k & 0 & \ddots & \vdots \\ \vdots & & & \ddots & b_{k+1} & a_{k+1} & b_{k+2} & \ddots & 0 \\ \vdots & & & & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & & & \ddots & \ddots & \ddots & b_n \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & b_n & a_n \end{bmatrix}$$

and P_{k+1} has the form

$$A_{k+1} = \left[\begin{array}{c|cc|c} I_{k-1} & \multicolumn{2}{c|}{O} & O \\ \hline & c_{k+1} & s_{k+1} & \\ O & & & O \\ & -s_{k+1} & c_{k+1} & \\ \hline O & \multicolumn{2}{c|}{O} & I_{n-k-1} \end{array}\right] \leftarrow \textbf{\textit{row}} \qquad (5.4)$$

↑ ***Column k***

where O denotes the appropriately dimensional matrix with all zero elements.

The sequences

$c_{k+1} = \cos\varphi_{k+1}$ and $s_{k+1} = \sin\varphi_{k+1}$ in P_{k+1} are chosen so that the (k+1, k) element in $A_{k+1}^{(1)}$ is zero; that is, $s_{k+1}x_k - c_{k+1}b_{k+1} = 0$. Since $c_{k+1}^2 + s_{k+1}^2 = 1$, the solutions of this equation are given by

$$s_{k+1} = \frac{b_{k+1}}{\sqrt{b_{k+1}^2 + x_k^2}} \quad and \quad c_{k+1} = \frac{x_k}{\sqrt{b_{k+1}^2 + x_k^2}},$$

and A $_{k+1}^{(1)}$ has the following form

$$A_k^{(1)}=\begin{bmatrix} z_1 & q_1 & r_1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & & & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & & & \vdots \\ \vdots & \ddots & 0 & z_k & q_k & r_k & \ddots & & \vdots \\ \vdots & & \ddots & 0 & x_{k+1} & y_{k+1} & 0 & \ddots & \vdots \\ \vdots & & & \ddots & b_{k+2} & a_{k+2} & b_{k+3} & \ddots & 0 \\ \vdots & & & & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & & & \ddots & \ddots & \ddots & b_n \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & b_n & a_n \end{bmatrix}$$

The process is repeated with this construction in the sequence P_2, …, Pn produces the upper-tridiagonal matrix

$$R^{(1)} \equiv A_k^{(1)} = \begin{bmatrix} z_1 & q_1 & r_1 & 0 & . & 0 \\ 0 & . & . & . & . & . \\ . & . & . & . & . & 0 \\ . & & . & . & . & r_{n-2} \\ . & & & . & z_{n-1} & q_{n-1} \\ 0 & . & . & . & 0 & x_n \end{bmatrix}$$

It remains the factorization of the matrix

$$Q^{(1)} = P_2^t P_3^t \dots\ P_n^t ,$$

using the QR – method.

Since the orthogonality of the rotation matrices implies that

$$Q^{(1)} R^{(1)} = (P_2^t P_3^t \dots\ P_n^t)\,(P_2^t P_3^t \dots\ P_n^t)$$
$$= A^{(1)}$$

The matrix $Q^{(1)}$ is orthogonal since

$$(Q^{(1)})^t Q^{(1)} = (P_2^t P_3^t \dots . P_n^t).(P_2^t P_3^t \dots . P_n^t)$$
$$= (P_n \dots P_3 P_2).(P_2^t P_3^t \dots P_n^t) = I$$

where, $Q^{(1)}$ is an upper Hessenberg matrix. Therefore, $A^{(1)} = R^{(1)}Q^{(1)}$ is also an upper Hessenberg matrix. Multiplying $Q^{(1)}$ on the left by the upper-triangular matrix $R^{(1)}$ does not affect the elements in the lower triangle. This implies that $A^{(2)}$ is in tridiagonal form, since it is symmetric.

The elements off the diagonal of $A^{(2)}$ will generally be smaller in absolute value than the corresponding elements of $A^{(1)}$, so $A^{(2)}$ is closer to being a diagonal matrix than is $A^{(1)}$. The process is repeated until $A^{(3)}$, $A^{(4)}$,… are constructed.

5.1. Acceleration Shift Technique

Although, the QR - method works much faster on special matrices such as symmetric tridiagonal matrices, Hessenberg matrices, and symmetric band matrices, comparing with other types of matrices, but convergence is still slow even for matrices of small dimension. Adding shifting technique speeds up the rate of convergence [1] and [6].

Assume that the eigenvalues of A are $\lambda_1, \lambda_2, \ldots, \lambda_n$,

where $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|$. The diagonalization process is started with this matrix.

The rate of convergence of the element $b_{j+1}^{(i+1)}$ to zero in the matrix $A^{(i+1)}$ depends on the ratio $|\lambda_{j+1}/\lambda_j|$. The rate of convergence of $b_{j+1}^{(i+1)}$ to zero determines the rate at which the element $a_j^{(i+1)}$ converges to the j^{th} eigenvalue λ_j. Thus, the rate of convergence can be slow if the ratio $|\lambda_{j+1}/\lambda_j| \approx 1$. In order to accelerate the rate of convergence a shifting technique will be used as follows.

A constant σ is selected near to an eigenvalue of A. This idea is incorporated in the modified the factorization in equation (5.3) to choosing $Q^{(i)}$ and $R^{(i)}$, so that

$$A^{(i)} - \sigma I = Q^{(i)} R^{(i)} \qquad (5.5)$$

then form

$$A^{(i+1)} = R^{(i)} Q^{(i)} + \sigma I \qquad (5.6)$$

This modification implies that, the rate of convergence of $b_{j+1}^{(i+1)}$ to zero depends on the ratio $\left|(\lambda_{j+1} - \sigma)/(\lambda_j - \sigma)\right|$.

If σ is chosen so that $\sigma \approx \lambda_{j+1}$, but $\sigma \not\approx \lambda_j$ then the original rate of convergence of $a_j^{(i+1)}$ to λ_j is determined.

The value σ in equation (5.5) is changed at each iteration so that when A has distinct eigenvalues then $b_{j+1}^{(i+1)}$ converges to zero faster than $b_j^{(i+1)}$ for any integer j < n.

Let $\lambda_n \approx a_n^{(i+1)}$ and $b_n^{(i+1)}$ is sufficiently small, delete the n^th row and column of the matrix, and continue in the same way to find an approximation to λ_{n-1}. Then QR iterating with shifting is repeated until an approximation has been found for each eigenvalue.

The shifting technique chooses at the i^th iteration, the shifting constant σ_i, where σ_i is the eigenvalue of the matrix

$$E^{(i)} = \begin{bmatrix} a_{n-1}^{(i)} & b_n^{(i)} \\ b_n^{(i)} & a_n^{(i)} \end{bmatrix}$$

that is closest to $a_n^{(i)}$. This shift translates the eigenvalues of A by a factor σ_i.

The method collects these shifts until $b_n^{(i+1)} \approx 0$ and then adds the shifts to $a_n^{(i+1)}$ for approximation of the eigenvalue λ_n.

If A has eigenvalues of the same absolute value then $b_j^{(i+1)}$ approaches zero for some $j \neq n$ at a faster rate than $b_n^{(i+1)}$.

Successive iteration is applied to smaller pair of submatrices obtained by the matrix splitting technique is described in (5.2).

6. QR Algorithm

There are several algorithms [3] treating the QR - method for finding the eigenvalues of the symmetric tridiagonal n × n matrix

$$A = A^{(1)} = \begin{bmatrix} a_1^{(1)} & b_2^{(1)} & 0 & . & 0 \\ b_2^{(1)} & a_2^{(1)} & . & & . \\ 0 & . & . & . & 0 \\ . & & & & \\ & . & . & . & b_n^{(1)} \\ . & & & & \\ 0 & .. & 0 & b_n^{(1)} & a_n^{(1)} \end{bmatrix}.$$

The QR algorithm (for more details see [1] and [3]) that we shall present can be found in and. The Pseudo-code of the algorithm is as in the following.

- Set dimension n; $a_1^{(1)},...,a_n^{(1)},b_2^{(1)},...,b_n^{(1)}$;

  tol; maximum number of iterations *m*.
- k = 1;

% collected shift
- *shift* = 0;
- while k ≤ m do
- if $\left|b_2^{(k)}\right| \le tol$ then

  $\lambda = a_n^{(k)} + shift;$

  display (λ);

  n = n-1;

  if $\left|b_2^{(k)}\right| \le tol$ then

    $\lambda = a_n^{(k)} + shift;$

    display (λ);

    n = n-1;

    $a_1^{(k)} = a_2^{(k)};$

for j = 2, … ,n

$a_j^{(k)} = a_{j+1}^{(k)}; \; b_j^{(k)} = b_{j+1}^{(k)};$

- if n = 0 then pause
- if n = 1 then $\lambda = a_1^{(k)} + shift;$

  display(λ);

  pause.
- for j = 3, … ,n-1

  if $\left|b_2^{(k)}\right| \le tol$ then display ('split into',

  $a_1^{(k)},\ldots,a_{j-1}^{(k)},b_2^{(k)},\ldots,b_{j-1}^{(k)},$ 'and', $a_j^{(k)},\ldots,a_n^{(k)},b_{j+1}^{(k)},\ldots,b_n^{(k)},shift);$

  pause

% compute shift

- $b = -(a_{n-1}^{(k)} + a_n^{(k)});$

  $c = a_n^{(k)}a_{n-1}^{(k)} - [b_n^{(k)}]^2;$

  $d = (b^2 - 4c)^{1/2};$
- if b > 0 then $\mu_1 = -2c/(b+d);$

  $\mu_2 = -(b+d)/2;$

  else $\mu_1 = (d-b)/2;$

  $\mu_2 = 2c/(d-b);$
- if n = 2 then $\lambda_1 = \mu_1 + shift;$

  $\lambda_2 = \mu_2 + shift;$

  display(λ_1, λ_2);

  pause.

% choose σ so that

- $\left|\sigma - a_n^{(k)}\right| = \min.\left\{ \left|\mu_1 - a_n^{(k)}\right|, \; \left|\mu_2 - a_n^{(k)}\right| \right\};$

% accumulate the shift

- $shift + shift + \sigma;$

% perform shift

- for j = 1, … ,n

  $d_j = a_j^{(k)} - \sigma$;

% compute $R^{(k)}$

- $x_1 = d_1; \ \ y_1 = b_2$;
- for j = 2, … ,n

  $z_{j-1} = sqrt[x^2{}_{j-1} + b_j{}^{(k)}]^2$

  $c_j = \dfrac{x_{j-1}}{z_{j-1}}; \ \ \sigma_j = \dfrac{b_j^{(k)}}{z_{j-1}}$;

  $q_{j-1} = c_j y_{j-1} + s_j d_j$;

  $x_j = -\sigma_j y_{j-1} + c_j d_j$;

  if $j \neq n$ then $r_{j-1} = \sigma_j b_{j+1}^{(k)}$;

  $y_j + c_j b_{j+1}^{(k)}$;

% $A_j^{(k)} = p_j A_{j-1}^{(k)}$ has just been computed

and $R^{(k)} = A^{(k)}$

% compute $A^{(k+1)}$

- $z_n = x_n$;

  $a_1^{(k+1)} = \sigma_2 q_1 + c_2 z_1; \ b_2^{(k+1)} = \sigma_2 z_2$;

- for j = 2, 3, …, n-1

  $a_j^{(k+1)} = \sigma_{j+1} q_j + c_j c_{j+1} z_j \ ; b_{j+1}^{(k+1)} = \sigma_{j+1} z_{j+1}$;

- $a_n^{(k+1)} = c_n z_n$;
- k = k + 1;

- End.

7. An Example Calculation

We shall illustrate the use of the QR algorithm described in the previous section for approximation of the eigenvalues of the symmetric tridiagonal 3 × 3 matrix

$$A = \begin{bmatrix} 3 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 3 \end{bmatrix}.$$

To follow the steps of the QR algorithm, for computing the eigenvalues of A, let $A = \begin{bmatrix} 3 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 3 \end{bmatrix} = \begin{bmatrix} a_1^{(1)} & b_2^{(1)} & 0 \\ b_2^{(1)} & a_2^{(1)} & b_3^{(1)} \\ 0 & b_3^{(1)} & a_3^{(1)} \end{bmatrix}$.

We have n = 3.

Let k = 1.

shift = 0.

b = - ($a_2^{(1)} + a_3^{(1)}$) = -6;

c = $a_3^{(1)} a_2^{(k)} - [b_3^{(1)}]^2$ = 8;

d = $(b^2 - 4c)^{0.5}$ = 2.

Since b < 0 so continuing the computation gives

$$\mu_1 = \frac{(d-b)}{2} = 4;$$

$$\mu_2 = \frac{2c}{(d-b)} = 2;$$

Choose σ so that

$$\left|\sigma - a_3^{(1)}\right| = \min.\left\{ \left|\mu_1 - a_3^{(1)}\right|,\ \left|\mu_2 - a_3^{(1)}\right| \right\}$$

$\sigma_1 = 2$;

$shift + shift + \sigma$ = 0 + 2 = 2;

Find $d_j = a_j^{(1)} - \sigma$, j = 1, 2, 3

$$A_1^{(1)} = \begin{bmatrix} d_1 & b_2^{(1)} & 0 \\ b_2^{(1)} & d_2 & b_3^{(1)} \\ 0 & b_3^{(1)} & d_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

Continuing the computation gives

$x_1 = d_1 = 1;\ \ y_1 = b_2^{(1)} = 1;$

$For\ \ j = 2,\ 3\ \ we\ \ have:$

$z_1 = \sqrt{2};\ \ c_2 = \dfrac{\sqrt{2}}{2};\ \ \sigma_2 = \dfrac{\sqrt{2}}{2};$

$q_1 = \sqrt{2};\ \ x_2 = 0;\ \ ;\ \ \text{since}\ \ j \neq n\ \ so$

$r_1 = \dfrac{\sqrt{2}}{2};\ \ and\ \ y_2 = \dfrac{\sqrt{2}}{2};$

$$\therefore\ \ A_2^{(1)} = P_1 A_1^{(1)} = \begin{bmatrix} \sqrt{2} & \sqrt{2} & \dfrac{\sqrt{2}}{2} \\ 0 & 0 & \sqrt{2} \\ 0 & 1 & 1 \end{bmatrix}.$$

Further,

$z_2 = 1;\ \ c_3 = 0;\ \ \sigma_3 = 1;$

$q_2 = 1;\ \ and;\ \ x_2 = \dfrac{-\sqrt{2}}{2};$

$$\text{so}\ \ R^{(1)} = A_3^{(1)} = \begin{bmatrix} \sqrt{2} & \sqrt{2} & \dfrac{\sqrt{2}}{2} \\ 0 & 1 & 1 \\ 0 & 0 & -\dfrac{\sqrt{2}}{2} \end{bmatrix}$$

In order to compute $A^{(2)}$, we have

$z_3 = \dfrac{-\sqrt{2}}{2};\ \ a_1^{(2)} = 2\ \ ;\ \ b_2^{(2)} = \dfrac{\sqrt{2}}{2};$

$For\ \ j = 2$

$a_2^{(2)} = 1;\ \ b_3^{(2)} = \dfrac{-\sqrt{2}}{2}; and\ \ a_3^{(2)} = 0.$

$$\therefore\ A^{(2)} = R^{(1)}Q^{(1)} = \begin{bmatrix} 2 & \frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & 1 & -\frac{\sqrt{2}}{2} \\ 0 & -\frac{\sqrt{2}}{2} & 0 \end{bmatrix}.$$

The first iteration of the QR - method is completed. Since neither $b_2^{(2)} = \frac{\sqrt{2}}{2}$, nor $b_3^{(2)} = \frac{-\sqrt{2}}{2}$ is small, iteration two of the QR - method is performed as follows.

Let k = 2.

shift = 0.

b = - ($a_2^{(2)} + a_3^{(2)}$) = -1;

c = $a_3^{(2)} a_2^{(2)} - [b_3^{(2)}]^2$ = - 0.5;

d = $(b^2 - 4c)^{0.5} = \sqrt{3}$.

Since b < 0 so continuing the computation gives

$$\mu_{1,2} = \frac{1}{2} \pm \frac{1}{2}\sqrt{3}$$

Choose σ so that

$$\left|\sigma - a_3^{(2)}\right| = \min\left\{\left|\mu_1 - a_3^{(2)}\right|, \left|\mu_2 - a_3^{(2)}\right|\right\}$$

$$\sigma_1 = \frac{1}{2} - \frac{1}{2}\sqrt{3}$$

Find $d_j = a_j^{(2)} - \sigma$, j = 2, 3

$$A_1^{(2)} = \begin{bmatrix} 2.3660 & 0.7071 & 0.0000 \\ 0.7071 & 1.3660 & -0.7071 \\ 0.0000 & -0.7071 & 0.3660 \end{bmatrix}$$

$x_1 = d_1 = 2.3660; \;\; y_1 = b_2^{(2)} = 0.7071;$

$For \;\; j = 2,\; 3 \;\; we \;\; have:$

$z_1 = 2.4694; \;\; c_2 = 0.9581; \;\; \sigma_2 = 0.2863; \qquad \therefore \qquad A_2^{(2)} =$

$q_1 = 1.1063; \;\; x_2 = 1.1063; \;\; ; \;\; \sin ce \;\; j \neq n \;\; so$

$r_1 = 0.2025; \;\; and \;\; y_2 = -0.6775;$

$$\begin{bmatrix} 2.4694 & 1.0687 & -0.2025 \\ 0.0000 & 0.0000 & 1.0687 \\ 0.0000 & -0.7071 & 0.3660 \end{bmatrix}.$$

Further,

$z_2 = 1.3130; \;\; c_3 = 0.8426; \;\; \sigma_3 = -0.5385;$

$q_2 = -0.7698; \;\; and; \;\; x_2 = -0.05633;$

$$\therefore \; R^{(2)} \equiv A_3^{(2)} = \begin{bmatrix} 2.4694 & 1.0687 & -0.2025 \\ 0.0000 & 1.3130 & -0.7698 \\ 0.0000 & 0.0000 & -0.0563 \end{bmatrix}$$

Computing A$^{(3)}$ is required to find

$z_3 = x_3 = -0.0563; \;\; a^2{}_1 = 2.6720; \;\; b^{(2)}{}_2 = 0.3759.$

$For \;\; j = 2,\; 3$

$a^{(3)}{}_2 = 1.4736; \;\; b^{(3)}{}_3 = 0.0304; \;\; a^{(2)}{}_3 = -0.0476;$

$$A^{(3)} = \begin{bmatrix} 2.6720 & 0.3759 & 0.0000 \\ 0.3759 & 1.4736 & 0.0304 \\ 0.0000 & 0.0304 & -0.0746 \end{bmatrix}.$$

If $b_3^{(3)} = 0.0304$ is sufficiently small, then the approximation to the eigenvalue λ_3 is 1.5864, the sum of $a_3^{(3)} = -0.0476$ and the shift $\sigma_1 + \sigma_2 = 2 + \frac{(1-\sqrt{3})}{2}$. Deleting the third row and column gives

$$A^{(3)} = \begin{bmatrix} 2.6720 & 0.3759 \\ 0.3759 & 1.4736 \end{bmatrix},$$

which has eigenvalues $\mu_1 = 2.7802$ and $\mu_2 = 1.3654$. Adding the shifts gives the approximations $\lambda_1 \approx 4.4142$ and $\lambda_2 \approx 2.9994$. Since the actual eigenvalues of the matrix A are 4.4142, 3.0000, and 1.5858, then the QR - method gave the approximation to the eigenvalues in two iterations only.

8. Computer Implementation and Results

8.1. Implementation

The above described two algorithms (Householder and QR algorithm) were implemented in Matlab software programming language. The two Matlab functions, namely ("***Program 1***" & "***Program 2***") in the Appendix are written as a function M - files (see [5] for details on Matlab programming language and its availability) ***Householder.m*** and ***QR_method_shift.m***. The Matlab ***Program 1*** shows a direct implementation of the Householder algorithm and can be used to reduce a real symmetric matrix A to a similar tridiagonal matrix. The Matlab ***Program 2*** uses the QR algorithm with acceleration shifts to approximate all the eigenvalues of the real symmetric tridiagonal matrix obtained by Matlab ***Program 1***. The function M-file ***QR_method_shift.m*** (Program ***2***) called by the function M-file ***Householder.m*** (***Program 1***). The Matlab ***Program 2*** follows from the QR algorithm, but with the following exceptions:

- The Matlab command ***eig*** is used to approximate the eigenvalues of the matrix

$$E^{(i)} = \begin{bmatrix} a_{n-1}^{(i)} & b_n^{(i)} \\ b_n^{(i)} & a_n^{(i)} \end{bmatrix}.$$

- The QR factorization of the matrix

$A^{(i)} - \sigma I = Q^{(i)} R^{(i)}$ is executed using the Matlab command ***[Q,R] = qr(E)***. This command produces an orthogonal matrix $Q^{(i)}$ and upper-triangular matrix $R^{(i)}$, such that

$E^{(i)} = Q^{(i)}R^{(i)}$.

8.2. Results

Running the Householder and QR algorithm programs ("***Program 1***" & "***Program 2***") from the Appendix on this input gives the following results.

1. The original (n x n) symmetric matrix is:

A=A^(1) =

4	1	-2	2
1	2	0	1
-2	0	3	-2
2	1	-2	-1

2. The symmetric tridiagonal matrix A^(n-1) similar to A = A^(1) using Householder algorithm is:

A^(n-1) =

4.0000	-3.0000	0	0
-3.0000	3.3333	-1.6667	0
0	-1.6667	-1.3200	0.9067
0	0	0.9067	1.9867

3. The eigenvalues of the symmetric tridiagonal (n×n) matrix A^(n-1) using

QR-method with acceleration shift is:

6.8446
-2.1975
1.0844
2.2685.

9. Conclusions

Given a real symmetric matrix A, an eigenvalues of A can be approximated by the QR – method after application of the Householder's method which reduces A to tridiagonal form. Shifting techniques aided in computing eigenvalues of the matrix A for accelerating the rate of convergence. Two algorithms were given to efficiently utilize the approximation of the eigenvalues of the matrix A, namely Householder and QR algorithms are those describe Householder's and QR – methods. Examples have been rendered which illustrates the two algorithms based on the two methods. A Matlab implementations of the Householder algorithm and QR algorithm which are coded as a Matlab functions (***Householder.m*** and ***QR_method_shift.m***) implement those algorithms. The results obtained by executing the Matlab functions. This results shows that there is some accuracy and performance improvement for approximation of eigenvalues of the real symmetric matrix A when going from the numerical procedure solution obtained without aid of computer to the fast computer procedure solution obtained either by the Householder's method, or by the QR – method.

Appendix

Program 1

This Matlab program implements the Householder reduction of (n x n) symmetric matrix to symmetric tridiagonal form.

```
function H=Householder (A)
clc; disp (' ');
A=[4 1 -2 2;1 2 0 1;-2 0 3 -2;2 1 -2 -1];
disp( '1. The original (n x n) symmetric matrix is:'); disp(' '); disp( ' A=A^(1)=');disp(' '); disp(A);
[m,n]=size(A);
m=n;
% Construct (n-2) Householder
  transformations.
for k=1: n-2
  q=0;
  for j=k+1: n
    q=q+A (j, k) ^2;
  end
  % Compute alpha
  if A (k+1, k)==0
    alpha=-sqrt (q);
  else
    alpha=
  (-sqrt (q)*A (k+1, k))/ (norm (A (k+1, k)));
  end
  mrs=alpha^2-alpha*A (k+1, k);
  % Notice that mrs=2*r^2
  % Construct v
  v (k) =0;
  v (k+1) =A (k+1, k)-alpha;
  for j=k+2: n
    v (j) =A (j, k);
  end
  % Construct u
  for j=k: n
```

```
      u (j) =0;
      for i=k+1: n
        u (j) =u (j) +A (j, i)*v (i);
      end
      u (j) =u (j)/mrs;
    end
    mult=0;
    for i=k+1: n
      mult=mult+v (i)*u (i);
    end
    for j=k: n
      z (j) =u (j)-(mult/ (2*mrs))*v (j);
    end
    % Construct the matrices A(2), A(3),
        A(4), ..., A(n-1).
    for l=k+1: n-1
      for j=l+1: n
        A (j, l) =A (j, l)-v (l)*z (j)-v (j)*z (l);
        A (l, j) =A (j, l);
      end
      A(l,l)=A(l,l)-2*v(l)*z(l);
    end
    A (n, n) =A (n, n)-2*v (n)*z (n);
    for j=k+2: n
      A (k, j) =0; A (j, k) =0;
    end
    A (k+1, k) =A (k+1, k)-v (k+1)*z (k);
    A (k, k+1) =A (k+1, k);
end
disp(' ');
disp( ' 2. The symmetric tridiagonal matrix A^(n-1) similar to A = A^(1) using
    Householder algorithm is:');disp(' ');
disp( '    A^(n-1) =');disp(' ');
disp(A);
epsilon=eps;
```

```
QR_method_shift(A,epsilon);
```

Program 2

This Matlab program implements the QR- method with acceleration shifts for computation the eigenvalues of the symmetric tridiagonal (n x n) matrix $A^{(n-1)}$ obtained by the Householder's method.

```
function D=QRMWAS(A,epsilon)
[m,n]=size(A);
m=n;
D=zeros(n,1);
E=A;
while (m>1)
   while (abs(E(m,m-1))>=epsilon)
      % calculate shift.
      sigma=eig(E(m-1:m,m-1:m));
      [i,j]=min([abs(E(m,m)*[1,1]'-sigma)]);
      % QR factorization of E.
      [Q,R]=qr(E-sigma(j)*eye(m));
      % Calculate next E.
      E=R*Q+ sigma(j)*eye(m);
   end
   % Place mth eigenvalue in A(m,m).
   A(1:m,1:m)=E;
   % Repeat process on the (m-1)x(m-1)
      submatrix of A.
   m=m-1;
   E=A(1:m,1:m);
end
m=n;
disp(' ');
disp(' 3. The eigenvalues of the symmetric
    tridiagonal (n×n) matrix A^(n-1) using
    QR-method with acceleration shift is:');disp(' ');
disp(diag(A))
```

References

[1] Burden, R. L., and Faires, J. D., "Numerical Analysis" Eight Edition, Thomson Brooks Kole (2005).

[2] Burden, R.L., and Fairs, J.D. "Numerical Analysis" Third Edition, PWS-KENT Publishing Company (1985).

[3] Fröberg, Carl-EriK, "Numerical Mathematics - Theory and Computer Applications", The Benjamin/ Cummings Publishing Company, Inc. (1985).

[4] Golub, G. H., and Van Loan, C. F., "Matrix Computations" , Third Edition, The John Hopkins University Press, Baltimore (1996).

[5] Hahan, B.D., "Essential Matlab for Scientists and Engineers", Arnold, a Member of the Holder Heading Group (1997).

[6] Mathews, J., H., and Fink, K. D., "Numerical Methods Using Matlab", Third Edition, Prentice Hall (1999).

[7] Web site: http://math.fullerton.edu