5-1979

The QR Algorithm

Hsiao-yin Edith Chu
Utah State University

THE QR ALGORITHM

by

Hsiao-yin Edith Chu

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Mathematics

Approved:

UTAH STATE UNIVERSITY
Logan, Utah

1979

ACKNOWLEDGMENTS

I sincerely appreciate professor David S. Watkins for his assistance and criticism on this paper.

Also, I would like to extend my thanks to all members of the Mathematics Department, who made us foreign students feel at home.

Edith Chu

TABLE OF CONTENTS

CHAPTER I

THE POWER AND INVERSE POWER METHOD

In this section, we will consider two methods for computing an eigenvector and in addition the associated eigenvalue of a matrix A.

The Power Method

Let $A \varepsilon R^{nxn}$ be simple and have $\lambda_1, \ldots, \lambda_n$ as the eigenvalues, where $|\lambda_1|>|\lambda_2|\geqslant----\geqslant|\lambda_n|$. Consequently, the corresponding eigenvectors $x_1,\ldots,x_n$ form a basis of R^n. For any vector $q_o \varepsilon R^n$, we have

$$q_o=r_1x_1 + r_2x_2 + \ldots + r_nx_n.$$

Compute $Aq_o = r_1\lambda_1x_1 + r_2\lambda_2x_2 + \ldots + r_n\lambda_nx_n$, and let

$$q_1 = Aq_o/\lambda_1 = r_1x_1 + r_2(\lambda_2/\lambda_1)x_2 + \ldots + r_n(\lambda_n/\lambda_1)x_n.$$

Then compute Aq_1, let $q_2 = Aq_1/\lambda_1$, ... Repeat the same procedure like the following:

$$q_{j+1} = Aq_1/\lambda_1 = r_1x_1 + r_2(\lambda_2/\lambda_1)^{j+1}\ x_2 + \ldots$$
$$+ r_n(\lambda_n/\lambda_1)^{j+1}\ x_n.$$

Generally, we do not know λ_1 when we do the iteration, so we had better use some other convenient scaling factor, say σ_j at each step, instead of λ_1. Thus our recursive formula will become

$$q_{j+1} = Aq_j/\sigma_j = \beta_1^{(j+1)}x_1+\beta_2^{(j+1)}(\lambda_2/\lambda_1)^{j+1}x_2 + \dots$$
$$+ \beta_n^{(j+1)}(\lambda_n/\lambda_1)^{j+1}x_n. \qquad (1.1)$$

where we choose $\sigma_j = \|Aq_j\|_\infty$, so that $\|q_{j+1}\|_\infty = 1$.

Since $|\lambda_1|>|\lambda_2|\geqslant\text{----}>|\lambda_n|$, this implies

$$1>|\lambda_2/\lambda_1|\geqslant|\lambda_3/\lambda_1|\geqslant\text{----}\geqslant|\lambda_n/\lambda_1|,$$

and thus the dominant ratio term in (1.1), $|\lambda_2/\lambda_1|^{j+1}$ converges to zero as $j\to\infty$. So that q_{j+1} will converge to β_1x_1, a multiple of x_1, with $\|\beta_1x_1\|_\infty = 1$, and the rate of convergence is $|\lambda_2/\lambda_1|$ when $j\to\infty$. In addition, since

$$(Aq_j,q_{j+1}) = (\sigma_jq_{j+1},q_{j+1}) = \sigma_j \text{ and } q_j,$$

q_{j+1} will both converge to β_1x_1 as $j\to\infty$, thus σ_j converges to λ_1 as $j\to\infty$.

Practical computation

From the above discussion, we know the approximate eigenvector will be scaled with norm of unity. Thus to choose $q_o = \begin{bmatrix}1\\1\end{bmatrix}$ as our initial guess is reasonable.

We will see some simple examples, and let q_j^1, q_j^2 be the two components of q_j, e_j be the error between the exact dominant eigenvalue and the approximate eigenvalue.

Example 1. $A=\begin{bmatrix} 1 & 4 \\ 7 & 2 \end{bmatrix}$

j	σ_j	q_j^1	q_j^2	e_j
2	5.8888889	.7735849	1.0000000	.9261840
4	6.5063613	.7137270	1.0000000	.3087116
6	6.7163620	.6958776	1.0000000	.0987109
8	6.7839409	.6903687	1.0000000	.0311320
10	6.8052972	.6886505	1.0000000	.0097757
12	6.8120075	.6881128	1.0000000	.0030654
14	6.8141121	.6879444	1.0000000	.0009608
16	6.8147718	.6878916	1.0000000	.0003011
18	6.8149785	.6878751	1.0000000	.0000944
20	6.8150433	.6878699	1.0000000	.0000296
22	6.8150636	.6878683	1.0000000	.0000093
24	6.8150700	.6878678	1.0000000	.0000029
26	6.8150720	.6878676	1.0000000	.0000009
28	6.8150726	.6878676	1.0000000	.0000003

The exact eigenvalues of A are $\lambda_1 = \frac{3+\sqrt{113}}{2} \simeq 6.8150729$ and $\lambda_2 = \frac{3-\sqrt{113}}{2} \simeq -3.8150729$, where $|\lambda_2/\lambda_1| = .5597$ $\simeq e_j/e_{j-1}$. One can see from Example 1, σ_j will converge to the dominant eigenvalue with rate of convergence $|\lambda_2/\lambda_1|$, and the scaled eigenvector associated with this eigenvalue will converge to $\begin{bmatrix} .6878676 \\ 1 \end{bmatrix}$.

Sometimes the approximate eigenvalue and eigenvector are unstable for a number of iterations, and the error is terrible. However, after a while, those values are convergent with a very slow rate and eventually stable.

Remarks: The convergence is slow if λ_1 is not strongly dominant over the other eigenvalues, or this method will not even converge if A does not have a dominant eigenvalue.

The Inverse Power Method

Suppose that A has eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ which correspond to the linear independent eigenvectors $x_1, \ldots, x_n$. Let μ be a close approximation to λ_1, then the eigenvalues of $(\mu I-A)^{-1}$ are $(\mu-\lambda_1)^{-1}, (\mu-\lambda_2)^{-1}, \ldots, (\mu-\lambda_n)^{-1}$ corresponding to $x_1, x_2, \ldots, x_n$. If we apply the power method to the matrix $(\mu I-A)^{-1}$, we will get

$$q_{j+1}\to\beta_1 x_1 \text{ and } \sigma_j\to(\mu-\lambda_1)^{-1}, \text{ so that } \mu-1/\sigma_j\to\lambda_1 \text{ as } j\to\infty.$$

Where the convergence rate depends on the largest of the ratios

$$\left|\frac{\mu-\lambda_1}{\mu-\lambda_i}\right| \quad i=2, \ldots, n,$$

which is much smaller than that in the power method, because of the way we have chosen μ.

Example 2. $A = \begin{bmatrix} 1 & 4 \\ 7 & 2 \end{bmatrix}$ $q_o = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ $\mu = 6$.

j	$\mu-1/\sigma_j$	q_j^1	q_j^2	e_j
1	6.6666667	1.0000000	1.0000000	.1484062
2	6.8275862	.6666667	1.0000000	.0125133
3	6.8140351	.6896552	1.0000000	.0010378
4	6.8151591	.6877193	1.0000000	.0000862
5	6.8150657	.6878799	1.0000000	.0000072
6	6.8150735	.6878665	1.0000000	.0000006
7	6.8150729	.6878676	1.0000000	.0000000
			$\mu = -4$	
1	-3.0000000	1.0000000	1.0000000	.8150729
2	-4.1666666	1.0000000	-1.0000000	.3515938
3	-3.8153846	-.8333333	1.0000000	.0003117
4	-3.8150782	-.8307692	1.0000000	.0000053
5	-3.8150730	-.8307255	1.0000000	.0000001
6	-3.8150729	-.8307247	1.0000000	.0000000

This method converges much faster than the power method and the convergence ratio is $\left|\frac{\mu-\lambda_1}{\mu-\lambda_2}\right| \sim e_{j+1}/e_j$. One difficulty of this method is that we much choose a suitable initial guess μ before we can do the iteration. One good choice of μ is the Rayleigh quotient

$$\mu = \frac{q_o^T A q_o}{q_o^T q_o}$$

if we know the initial guess q_o is reasonable. Thus we can modify the inverse power method, and get Rayleigh quotient iteration by computing the Rayleigh quotient

$$\mu_j = \frac{q_j^T A q_j}{q_j^T q_j}$$

on each step. It will be discussed in detail later. Now we simply present some practical examples. The reader can compare these with the previous ones. It turns out that this iteration is quite good and converges very quickly.

Example 3. $A = \begin{bmatrix}1 & 4\\ 7 & 2\end{bmatrix}$ $q = \begin{bmatrix}1\\ 1\end{bmatrix}$

j	μ_j	q_j^1	q_j^2	e_j
1	6.8240000	.6923077	1.0000000	.0089271
2	6.8150805	.6878713	1.0000000	.0000076
3	6.8150729	.6878676	1.0000000	.0000000

Table 4. $B = \begin{bmatrix}9 & 1\\ 1 & 2\end{bmatrix}$ $q = \begin{bmatrix}1\\ 1\end{bmatrix}$

j	μ_j	q_j^1	q_j^2	e_j
1	8.0076923	1.0000000	-.2727273	1.1323626
2	9.0948443	1.0000000	.2215583	.0452107
3	9.1400532	1.0000000	.1395513	.0000018
4	9.1400549	1.0000000	.1400549	.0000000
		$q = \begin{bmatrix}1\\ .3\end{bmatrix}$		
1	9.1399597	1.0000000	.1363694	.0000952
2	9.1400549	1.0000000	.1400550	.0000000

The exact eigenvalues of B are $\lambda_1 = \frac{11+\sqrt{53}}{2} \simeq 9.1400549$ and $\lambda_2 = \frac{11-\sqrt{53}}{2} \simeq 1.8599451$.

CHAPTER II

ORTHOGONAL TRANSFORMATIONS

In this section, we will introduce two types of orthogonal transformation. These transformations will be used in this paper to reduce matrices.

Elementary Reflector (ER)

Definition

Let $x \varepsilon R^n$ with $\|x\|_2 = 1$, then $U=I-2xx^T$ is the Elementary Reflector (ER) corresponding to x.

Remark 1. If $x \varepsilon R^n$, $x \neq 0$ and $\|x\|_2 \neq 1$, we can extend the above definition by letting $x' = x/\|x\|_2$, then $\|x'\|_2 = 1$ and $U=I-2x'x'^T$ is the ER corresponding to x. So we can feel free with this definition for any nonzero vector.

Remark 2. For any $y \varepsilon R^n$, $y \neq 0$ and $y^T x = 0$ it is easy to show that $Ux=-x$ and $Uy=y$. The geometric interpretation is that the ER U corresponding to x simply reflects x itself to the opposite direction and leaves any vector orthogonal to x unchanged.

Theorem 2.1

Let U be an ER, then U is symmetric $(U=U^T)$, orthogonal $(U^T=U^{-1})$ and involutory $(U=U^{-1})$.

Proof. Since $U=I-2xx^T$

$$\begin{aligned} U^T &= (I-2xx^T)^T \\ &= I^T-(2xx^T)^T \\ &= I-2(xx^T)^T \\ &= I-2xx^T = U. \end{aligned}$$

$$\begin{aligned} U^TU &= UU \\ &= (I-2xx^T)(I-2xx^T) \\ &= I-2xx^T-2xx^T+4xx^Txx^T \quad \because \ \|x\|_2^2 = x^Tx = 1 \\ &= I-4xx^T+4xx^T = I \end{aligned}$$

Since $U=U^T$, $U^T=U^{-1}$ thus $U=U^{-1}$.

//

Theorem 2.2

Let $x,x'\varepsilon R^n$, $x\neq x'$ and $\|x\|_2 = \|x'\|_2$, then there exists an ER U such that $Ux = x'$.

Proof. Take $\hat{x} = \dfrac{x-x'}{\|x-x'\|_2}$ and let $U=I-2\hat{x}\hat{x}^T$, then

$$\begin{aligned} Ux &= U(\tfrac{1}{2}(x+x') + \tfrac{1}{2}(x-x')) \\ &= \tfrac{1}{2}U(x+x') + \tfrac{1}{2}U(x-x') \end{aligned}$$

since $((x+x'),\hat{x}) = (x+x')^T \dfrac{x-x'}{\|x-x'\|_2}$

$$= \frac{x^Tx+x'^Tx-x^Tx'-x'^Tx'}{\|x-x'\|_2} = 0 \quad \because \ \|x\|_2 = \|x'\|_2 \quad x^Tx = x'^Tx'$$

Thus $(x+x')$ is orthogonal to $\hat{x}$. On the other hand, we know that $(x-x')$ is parallel to $\hat{x}$. Hence $Ux = \frac{1}{2}(x+x') + \frac{1}{2}(x'-x) = x'$.

//

This theorem gives us a way to introduce zeros into a vector or matrix. That is, if

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \neq 0$$

we can find an ER U such that Ux = x' where

$$x' = \begin{bmatrix} \sigma \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad \sigma = \pm\|x\|_2.$$

Plane Rotation (PR)

Let $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$, and $r = \frac{x_1}{\|x\|_2}$, $t = \frac{x_2}{\|x\|_2}$. Then

$$\begin{bmatrix} r & t \\ -t & r \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \|x\|_2 \\ 0 \end{bmatrix}$$

In geometry, the matrix $\begin{bmatrix} r & t \\ -t & r \end{bmatrix}$ acts on x, so that one of the components is annihilated. That is, we rotate the R^2 plane in order to set this vector x on the x-axis.

Definition

A matrix of the form

$$P_j = \begin{array}{c|cccccccc|}
 & & & & i & & j & & \\
\hline
 & 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\
 & 0 & 1 & 0\cdots & 0 & & 0 & & 0 \\
 & \vdots & \vdots & & \vdots & & \vdots & & \vdots \\
i & 0 & 0 & \cdots & r & \cdots & t & \cdots & \cdots \\
 & \vdots & \vdots & & \vdots & & \vdots & & \vdots \\
j & 0 & 0 & \cdots & -t & \cdots & r & \cdots & 0 \\
 & \vdots & \vdots & & \vdots & & \vdots & & \vdots \\
 & 0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \\
\hline
\end{array}$$

is called a Plane Rotation (PR) in the (i,j)-plane. It should be noted that (1) by applying P_{ij}, we can also introduce a zero in a nonzero vector in R^n, and (2) it follows from the definition of plane rotation, that P_{ij} also satisfies the orthogonality. Thus it is easy to get the inverse matrix.

Both transformations we defined above, have the following common and important properties:

1. The condition number of U (or P_{ij}) with respect to the 2-norm is $\chi(U) = \|U\|_2\|U^T\|_2 = 1$. This implies that U is perfectly conditioned.

2. Suppose after having computed U^TAU, we obtained an error $F = U^TEU$. Then $\|E\|_2 = \|F\|_2$ and $U^T(A+E)U = U^TAU+F$. In other words, a perturbation in the result can be accounted by a perturbation of the same size in the original problem.

The QR algorithm will involve only these two orthogonal transformations, and thus can be expected to be numerically stable.

CHAPTER III

DEFLATION OF MATRICES AND THE RESIDUAL

Let x be an eigenvector of $A \varepsilon R^{nxn}$ with $\|x\|_2 = 1$, corresponding to eigenvalue λ. Let U be an ER such that $Ux = e_1$. Then, since $U = U^{-1}$, we have $Ue_1 = x$. It follows that the first column of U is x. Thus $U = [x,V]$, where the columns of V are orthogonal to x. Now U^TAU

$$= [x,V]^T A[x,V]$$

$$= \begin{bmatrix} x^TAx & x^TAV \\ V^TAx & V^TAV \end{bmatrix}$$

$$= \begin{bmatrix} \lambda & x^TAV \\ \lambda V^Tx & V^TAV \end{bmatrix}$$

$$= \begin{bmatrix} \lambda & h \\ 0 & \bar{A} \end{bmatrix}$$

where $h = x^TAV$ and $\bar{A} = V^TAV$. This gives a matrix $\bar{A}$ of order one less than A which has all eigenvalues of A except λ.

In practice, if we use an approximate eigenvector x' with $\|x'\|_2 = 1$, we will have $[x',V]$ as the orthogonal matrix.

$$[x',V]^T A[x',V] = \begin{bmatrix} x'^T Ax' & h \\ g & \bar{A} \end{bmatrix} \tag{3.1}$$

where $g = V^T Ax$.

To compute the deflation, we must set $g = 0$. This leads to an error of size $\|g\|_2$ in the original matrix.

Definition

Suppose that x' is an approximate eigenvector corresponding to the approximate eigenvalue μ. Define the residual vector to be $r = Ax'-\mu x'$.

Thus the size of r in some sense measures the accuracy of x' and μ.

Theorem 3.1

For fixed x, $\|r\|_2$ is a minimum when $\mu = \dfrac{x^T Ax}{x^T x}$.

Proof

$$\begin{aligned} \|\gamma\|_2^2 &= \|Ax-\mu x\|_2^2 \\ &= (Ax-\mu x)^T(Ax-\mu x) \\ &= x^T x\mu^2-2x^T Ax\mu+x^T A^T Ax. \end{aligned}$$

Let $f(\mu) = (x^T x)\mu^2-(2x^T Ax)\mu+x^T A^T Ax$ and set $f'(\mu) = 0$, solving for μ, we get

$$\mu = \frac{x^T Ax}{x^T x} .$$

Since $f''(\mu) = 2x\ x>0$, therefore $f(\mu)$ has the minimum value. That is, $\|r\|_2^2$ is a minimum, hence $\|r\|_2$ is a minimum when

$$\mu = \frac{x^T Ax}{x^T x} .$$

//

Theorem 3.2

Consider g in (3.1) where $\|x'\|_2 = 1$. Then $\|g\|_2 = \|Ax' - (x'^T Ax')x'\|_2$.

Proof

$$\|Ax' - (x'^T Ax')x'\|_2 = \|[x',V]^T[Ax' - (x'^T Ax')x']\|_2$$

$$\left\|\begin{bmatrix} x'^T[Ax' - (x'^T Ax')x'] \\ V^T[Ax' - (x'^T Ax')x'] \end{bmatrix}\right\|_2$$

$$\left\|\begin{bmatrix} 0 \\ g - (x'^T Ax')V^T x' \end{bmatrix}\right\|_2$$

$$\left\|\begin{bmatrix} 0 \\ g \end{bmatrix}\right\|_2$$

$$\|g\|_2$$

Since $\|x'\|_2 = 1$ by the orthogonality of $[x',V]$.

Corollary

$$\|g\|_2 = \min \|r\|_2.$$

The corollary says that the error introduced by setting g to zero in the deflation process is the same size as the smallest residual that can be obtained from the approximate eigenvector x'.

CHAPTER IV

RAYLEIGH QUOTIENT ITERATION

As mentioned at the end of Chapter I, we can refine the inverse power method by changing the approximate eigenvalue μ from one iteration to the next, where

$$\mu = \frac{q_i^T A q_i}{q_i^T q_i}$$

is the Rayleigh Quotient associated with each iteration. Also, we have seen in Theorem 1 in Chapter III that the choice of μ as the Rayleigh Quotient is justified by the fact that this value minimizes the residual. Since this is a very nice iteration, it will be the basis of QR algorithm. We will discuss the convergence of it first. After this, we will introduce the QR algorithm and demonstrate the close relation between them.

We will let the approximate eigenvector q_i be scaled so that it has norm unity, and call it x at each iteration. Assume the iteration is converging to an eigenvector which corresponds to a simple eigenvalue μ, and μ is not very close to the rest of the eigenvalues of A, the Rayleigh Quotient Iteration for matrix A is the following:

1. Calculate $\mu = \frac{x^T Ax}{x^T x}$ from the present approximate eigenvector x.

2. Find the next approximate eigenvector x' by $(\mu I-A)x' = \sigma x$, where σ is a scaling factor.

Let [x,V] be the ER which is defined as in Chapter III. Let $y = [x,V]^T x$ and $y' = [x,V]^T x'$ then $y = e_1$. Thus y and y' are Rayleigh Quotient iterates of the matrix $[x,V]^T A[x,V] = \begin{bmatrix} \mu & h \\ g & \bar{A} \end{bmatrix}$, $g = V^T Ax$. Then $(\mu I-[x,V]^T A[x,V])y' = \sigma y$

$$\begin{bmatrix} 0 & -h \\ -g & \mu I-\bar{A} \end{bmatrix} y' = \sigma e_1$$

$$\begin{bmatrix} 0 & -h \\ -g & \mu I-\bar{A} \end{bmatrix} \begin{bmatrix} 1 \\ \bar{y} \end{bmatrix} = \sigma e_1 .$$

Since the first component of y' is not zero, we can choose σ such that the first component of y' is unity, and obtain

$$\bar{y} = [\mu I-\bar{A}]^{-1} g \tag{4.1}$$

$$\text{Let } r = \frac{\|\mu y-([x,V]^T A[x,V]y)\|_2}{\|y\|_2}$$

$$= \left\| \begin{bmatrix} 0 & -h \\ -g & \mu I-\bar{A} \end{bmatrix} e_1 \right\|_2$$

$$= \left\| \begin{bmatrix} 0 \\ g \end{bmatrix} \right\|_2$$

$$= \|g\|_2$$

This is a measure of the accuracy of y as an eigenvector of $[x,V]^T A[x,V]$. Likewise, if μ' denotes the Rayleigh Quotient for y', then we have

$$r' = \frac{\|\mu' y' - ([x,V]^T A[x,V])y'\|_2}{\|y'\|_2} \qquad (4.2)$$

Since y' need not have norm unity, we determine an ER, say $P = [y",V']$ such that μ' is the Rayleigh Quotient of the first column of P corresponding to $[x,V]^T A[x,V]$ and y" has norm unity. By Chapter III, $P^T[x,V]^T A[x,V]P = \begin{bmatrix} \mu' & h' \\ g' & \bar{A}' \end{bmatrix}$ where

$$g' = V'^T \begin{bmatrix} \mu & h \\ g & \bar{A} \end{bmatrix} y" \text{ and then } r' = \|g'\|_2.$$

We can determine P as follows:

Since $y' = \begin{bmatrix} 1 \\ \bar{y} \end{bmatrix}$ from the definition of ER, let

$$x = \begin{bmatrix} 1+\|y'\|_2 \\ \bar{y} \end{bmatrix}$$

and

$$\pi = \|y'\|_2(1+\|y'\|_2),$$

then the ER $P = I-\pi^{-1}xx^T$

$$P = I - \frac{1}{\|y'\|_2(1+\|y'\|_2)}\begin{bmatrix}1+\|y'\|_2\\ \bar{y}\end{bmatrix}[1+\|y'\|_2\bar{y}^T]$$

$$= I - \frac{1}{\|y'\|_2(1+\|y'\|_2)}\begin{bmatrix}(1+\|y'\|_2)^2 & (1+\|y'\|_2)\bar{y}^T\\ (1+\|y'\|_2)\bar{y} & \bar{y}\bar{y}^T\end{bmatrix}$$

$$= I - \frac{1}{\|y'\|_2}\begin{bmatrix}1+\|y'\|_2 & \bar{y}^T\\ \bar{y} & \dfrac{\bar{y}\bar{y}^T}{1+\|y'\|_2}\end{bmatrix} .$$

We will consider the case if g is small, then by (4.1) $\bar{y}$ will be small, so that we can neglect $\bar{y}\bar{y}^T$ terms. This gives

$$P \sim I - \frac{1}{\|y'\|_2}\begin{bmatrix}1+\|y'\|_2 & \bar{y}^T\\ \bar{y} & 0\end{bmatrix}$$

$$= \frac{1}{\|y'\|_2}\begin{bmatrix}-1 & -\bar{y}\\ -\bar{y} & \|y'\|_2 I\end{bmatrix}$$

$$g' = V'^T\begin{bmatrix}\mu & h\\ g & \bar{A}\end{bmatrix} y''$$

$$= \begin{bmatrix}\dfrac{-y}{\|y'\|_2} & I\end{bmatrix}\begin{bmatrix}\mu & h\\ g & \bar{A}\end{bmatrix}\begin{bmatrix}-1/\|y'\|_2\\ -\bar{y}/\|y'\|_2\end{bmatrix}$$

$$= \begin{bmatrix}(\dfrac{-\mu\bar{y}}{\|y'\|_2}+g) & (-\bar{y}h/\|y'\|_2+\bar{A}I)\end{bmatrix}\begin{bmatrix}-1/\|y'\|_2\\ -\bar{y}/\|y'\|_2\end{bmatrix}$$

$$= (\mu\bar{y}-\|y'\|_2 g + \bar{y}h\bar{y}-\|y'\|_2\bar{A}\bar{y})/\|y'\|_2^2$$

$$= (\mu\bar{y}-\mu\|y'\|_2\bar{y}+\|y'\|_2\bar{A}\bar{y}+\bar{y}h\bar{y}-\|y'\|_2\bar{A}\bar{y})/\|y'\|_2^2$$

$$= (\mu\bar{y}-\mu\|y'\|_2\bar{y}+\bar{y}h\bar{y})/\|y'\|_2^2$$

Thus $\|g'\|_2 \leqslant \|\mu\bar{y}-\mu\|y'\|_2\bar{y}+\bar{y}h\bar{y}\|_2/\|y'\|_2^2$

$$\leqslant (\mu|1-\|y'\|_2|\, \|[\mu I-A]^{-1}\|_2\|g\|_2 +$$

$$\|h\|_2\, \|[\mu I-A]^{-1}\|_2^2\|g\|_2^2)/\|y'\|_2^2$$

Therefore $r' \leqslant (\mu\cdot|1-\|y'\|_2|\, \|[\mu I-A]^{-1}\|_2 r + \|h\|_2\, \|[\mu I-A]^{-1}\|_2^2 r^2)/$

$$\|y'\|_2^2 \qquad (4.3)$$

Since $\|y'\|_2 = (1+\|\bar{y}\|_2^2)^{1/2}$, by Taylor's expansion we have

$$\|y'\|_2 = 1+1/2\|\bar{y}\|_2^2 + 0(\|\bar{y}\|_2^4) \sim 1+1/2\|\bar{y}\|_2^2.$$

That is

$$1-\|y'\|_2^2 \sim 1/2\|\bar{y}\|_2^2.$$

Thus inequality (4.3) becomes

$$r' < (\|h\|_2\, \|[\mu I-A]^{-1}\|_2^2 r^2 + \frac{\mu}{2}\|\bar{y}\|_2^2\cdot \|[\mu I-A]^{-1}\|_2 r)/\|y'\|_2^2$$

$$= (\|h\|_2\, \|[\mu I-A]^{-1}\|_2^2\cdot r^2)/\|y'\|_2^2+(\frac{\mu}{2}\, \|[\mu I-A]^{-1}\|_2^3\cdot r^3)/\|y'\|_2^2$$

$$\sim (\|h\|_2\, \|[\mu I-A]^{-1}\|_2^2\cdot r^2)/\|y'\|_2^2$$

Hence we have

$$r' < c\|h\|_2 r^2 \qquad (4.4)$$

where

$$c = \|[\mu I-A]^{-1}\|_2^2/\|y'\|_2^2$$

That is, the Rayleigh Quotient iteration will converge quadratically when it converges. If A is symmetric, then h=g. Thus (4.4) changes into $r' < c\|g\|_2 r^2 = c'r^3$. This implies the rate of convergence will be cubic.

CHAPTER V

QR DECOMPOSITION

Theorem 5.1

Let $A \varepsilon R^{nxn}$, then there exists an orthogonal matrix Q and upper triangular R, with positive diagonal elements, such that A = QR.

Proof

By induction on n. If n = 1, [a] = [1][a]. Assume the result holds for n = k-1. Let $A \varepsilon R^{nxn}$ and $a_1 = \begin{bmatrix} a_{11} \\ \vdots \\ a_{k1} \end{bmatrix}$ be the first column of A. If $a_{21} = \ldots = a_{k1} = 0$, we can let $Q_1 = I$. Otherwise, let Q_1 be the ER such that $Q_1 a_1 = -\sigma e_1$. Let $A_1 = Q_1 A$, then $A_1 = \left[\begin{array}{c|c} -\sigma & r \\ \hline 0 & \hat{R}_1 \end{array}\right]$, this implies $A = Q_1^T A_1 = Q_1 A_1$.

Since $\hat{A}_1 \varepsilon R^{(k-1)x(k-1)}$, by the induction hypothesis $\hat{A}_1 = \hat{Q}_1 \hat{R}_1$, where $\hat{Q}_1$ is orthogonal and $\hat{R}_1$ is upper triangular. Let

$$Q_2 = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & \hat{Q}_1 \end{array}\right] \qquad R = \left[\begin{array}{c|c} -\sigma & r \\ \hline 0 & \hat{R}_1 \end{array}\right]$$

(Here we can choose the sign of σ such that $-\sigma$ is positive), then Q is orthogonal and R is upper triangular and

$Q_2R = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & Q_1 \end{array}\right]\left[\begin{array}{c|c} -\sigma & r \\ \hline 0 & R_1 \end{array}\right] \quad \left[\begin{array}{c|c} -\sigma & r \\ \hline 0 & R_1 \end{array}\right] = A_1$. We have $A =$

$Q_1A_1 = Q_1Q_2R$, let $Q = Q_1Q_2$, then $A = QR$.

//

It should be noted that the proof of this theorem is constructive.

If A is nonsingular, then the decomposition is unique: Let $A = Q_1R_1 = Q_2R_2$, where Q_1, Q_2 are orthogonal and R_1, R_2 are upper triangular with positive nonzero diagonal entrices. Since $A^TA = R_1^TQ_1^TQ_1R_1 = R_1^TR_1 = R_2^TR_2$. By the uniqueness of the choleski decomposition of A^TA, we have $R_1 = R_2$ and $Q_1 = Q_2$. If A is singular, for example, $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ then we can have $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ or $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$ as the QR decomposition of A.

If we want to find the eigenvalues of a matrix A, consider solving for the characteristic polynomial of A. There is no finite process through which one can solve a general polynomial of degree $\geqslant 5$. In other words, there is no algorithm employing a finite number of steps that can be used to solve for all eigenvalues of a general matrix.

For this reason, in general, we have to use some iterative method to find all the eigenvalues of a matrix.

CHAPTER VI

THE EXPLICIT QR ALGORITHM

We will basically use the QR decomposition to define and develop the QR algorithm. This yields a sequence $\{A_k\}$ of matrices orthogonally similar to A_1. We may expect the limiting matrix to be triangular or quasi-triangular. From now on, assume $A_1 \varepsilon R^{nxn}$ and A_1 is nonsingular.

Algorithm 6.1

For k = 1, 2, ..., decompose A_k into $Q_k R_k$. Then form $A_{k+1} = R_k Q_k$.

Algorithm 6.2

Choose an "origin shift" x_k. For k = 1, 2, ..., decompose $A_k - x_k I$ into $Q_k R_k$. Then form $A_{k+1} = R_k Q_k + x_k I$ (that is $A_{k+1} = Q_k^T A_k Q_k$).

We now discuss these two algorithms in detail.

Theorem 6.1

Let $\tilde{Q}_k = Q_1 Q_2 \ldots Q_k$ and $\tilde{R}_k = R_k R_{k-1} \ldots R_1$, then $\tilde{Q}_k \tilde{R}_k = (A_1 - x_k I)(A_1 - X_{k-1} I) \ldots (A_1 - x_1 I)$.

Proof. By induction on k. k = 1 then $\tilde{Q}_1\tilde{R}_1 = Q_1R_1 = A_1-x_1I$. Assume this result is true for k-1. Since $(A_{k+1}-x_kI) = Q_k^T(A_1-x_kI)Q_k$,

$$\begin{aligned} R_k &= (A_{k+1}-x_kI)Q_k^T \\ &= \tilde{Q}_k^T(A_1-x_kI)\tilde{Q}_kQ_k^T \\ &= \tilde{Q}_k^T(Q_1-x_kI)\tilde{Q}_{k-1} \qquad \text{and} \end{aligned}$$

$$\begin{aligned} \tilde{Q}_k\tilde{R}_k &= \tilde{Q}_kR_k\tilde{R}_{k-1} \\ &= \tilde{Q}_k\tilde{Q}_k^T(A_1-x_kI)\tilde{Q}_{k-1}\tilde{R}_{k-1} \\ &= (A_1-x_kI)\tilde{Q}_{k-1}\tilde{R}_{k-1} \end{aligned}$$

By the induction hypothesis $Q_kR_k = (A_1-x_kI)(A_1-x_{k-1}I) \ldots (A_1-x_1I)$. //

In the above theorem, if we take $x_1=x_2= \ldots = x_k=0$, this gives algorithm 6.1. Then we have $A_1^k = \tilde{Q}_k\tilde{R}_k$ and $A_{k+1} = \tilde{Q}_k^TA_1\tilde{Q}_k$. (That is, the orthogonal matrix $\tilde{Q}_k$ in A_{k+1} comes from the QR decomposition of A_1^k.) $A_1^ke_1 = \tilde{Q}_k\tilde{R}_ke_1= \tilde{q}_1^{(k)}\tilde{r}_{11}^{(k)}$, thus $\tilde{q}_1^{(k)} = \dfrac{A_1^ke_1}{r_{11}^{(k)}}$. From Chapter III, we have

$$A_{k+1} = \begin{bmatrix} a_{11}^{(k+1)} & h^{(k+1)} \\ g^{(k+1)} & A_1^{(k+1)} \end{bmatrix}$$

where $\|g^{(k+1)}\|_2$ is the minimal residual of $\tilde{q}_1^{(k)}$ which is obtained by applying k steps of the power method to e_1. If A_1 has a dominant eigenvalue λ_1, then $\tilde{q}_1^{(k)}$ approaches an eigenvector of A_1, so that $\|g^{(k+1)}\|_2 \to 0$. Hence the subdiagonal elements of the first column of A_{k+1} will approach zero linearly. Meanwhile, under some conditions, all the rest of the subdiagonal elements are also decreasing relatively slowly at each iteration:

Lemma 6.1

A_1 is simple if there exists a nonsingular matrix χ such that $\chi^{-1}A_1\chi = \Lambda$, where Λ = diagonal $[\lambda_1, \ldots, \lambda_n]$.

Theorem 6.2

Let A_1 be simple and satisfy (1) $|\lambda_1|>|\lambda_2|> \ldots >|\lambda_n|$; (2) χ^{-1} has an LU decomposition, then the subdiagonal elements of the matrix A_{k+1} tend to zero.

Proof. See [2].

From algorithm 6.2 $(A_k - x_k I) = Q_k R_k$, take this equation, invert it and transpose it, then solve for Q_k. We will have $(A_k - x_k I)^{-1} = R_k^{-1} Q_k^{-1}$

$$(A_k - x_k I)^{-T} = Q_k^{-T} R_k^{-T}$$

$$(A_k^T - x_k I)^{-1} = Q_k^{-T} R_k^{-T}$$

$$(A_k^T - x_k I)^{-1} R_k^T = Q_k^{-T} = Q_k.$$

Then $q_n^{(k)} = Q_k e_n = r_{nn}(A_k^T - x_k I)^{-1} e_n$. That is, $q_n^{(k)}$ is the approximate eigenvector of A_k^T obtained by applying one step of the inverse power method to e_n.

It follows from Chapter III that we can partition A_k as

$$A_k = \begin{vmatrix} V^T A_k V & h^{T(k)} \\ g^{T(k)} & a_{nn}^{(k)} \end{vmatrix}$$

Then $\|g^{T(k)}\|_2 = \|g^{(k)}\|_2$ is the minimum residual of e_n regarded as an approximate eigenvector of A_k^T since $A_{k+1} = Q_k^T A_k Q_k$. $\|g^{(k+1)}\|_2$ is the minimum residual of $q_n^{(k)}$ regarded as an approximate eigenvector of A_k^T. If x_k is near an eigenvalue of A_k^T, then $q_n^{(k)}$ will be a more accurate approximate eigenvector than e_n. Thus $\|g^{(k+1)}\|_2$ is smaller than $\|g^{(k)}\|_2$. Theregore $g^{(k+1)} \to 0$ and we will get the approximate eigenvalue $a_{nn}^{(k)}$ corresponding to $q_n^{(k)}$ as $k \to \infty$.

Since we have discussed the very nice properties of Rayleigh Quotient iteration in Chapter IV, we now choose x_k as the Rayleigh Quotient at each step to get $x_k = e_n^T A_k^T e_n = a_{nn}^{(k)}$. Hence $g^{(k+1)} \to 0$ quadratically and $g^{(k+1)} \to 0$ cubically if A_1 is symmetric.

In fact, after ℓ iterations $(\ell < k)$, the shifts $\chi_{\ell+1}$, $\chi_{\ell+2}$, ... in algorithm 6.2 are very close to an eigenvalue λ_n. By theorem 6.1 we get $\tilde{Q}_k \tilde{R}_k = (A_1 = x_1 I)_1 (A_1 - x_2 I)_2 \ldots (A_1 - x_\ell I)\ (A_1 - \lambda_n I)^{k-\ell}$. Thus only the previous ℓ iterations are obtained by using the shifted algorithm on A_1, and the rest of the steps are achieved by applying the unshifted algorithm on $A_1-\lambda_n I$. It follows from Theorem 6.2 that the shifted algorithm will not only reduce the off-diagonal elements in the last row to zero, but may simultaneously reduce the other subdiagonal elements, somewhat more slowly.

CHAPTER VII

HESSENBERG FORM

Definition

A matrix A is said to be an upper Hessenberg matrix ($A \varepsilon UHM$) if $a_{ij} = 0$ for $i>j+1$. If also $a_{i+1,i} \neq 0$ for $i=1, \ldots, n-1$, then we say A is unreduced ($A \varepsilon UUHM$)

Theorem 7.1

Any matrix $A \varepsilon R^{nxn}$ can be transformed by ER's $\hat{U}_{n-2}, \hat{U}_{n-3}, \ldots, \hat{U}_1$, so that $A_1 = \hat{U}_{n-2} \hat{U}_{n-3} \ldots \hat{U}_1 A \hat{U}_1 \hat{U}_2 \ldots \hat{U}_{n-2}$ is an upper Hessenberg matrix ($A_1 \varepsilon UHM$).

We will not prove the Theorem. Instead, we will later see Example 7.1 for the practical detail.

If any subdiagonal element of A_1 vanishes, then A_1 is said to be reduced and we can partition A_1 as follows:

$$A_1 = \begin{bmatrix} H_1 H_{12} & \cdots & H_{1m} \\ H_2 & \ddots & \vdots \\ 0 & & H_m \end{bmatrix} \qquad (7.1)$$

where $H_i \varepsilon UUHM$.

Since A_1 is similar to A, we can solve for all eigenvalues of A_1 in order to get those of A. We control the computing program so that the QR algorithm acts independently on each H_i.

Remarks. (1) If any H_i in (7.1) is a 1x1 block, then this element is one eigenvalue of A. (2) For any $H_i \varepsilon R^{2x2}$, we can solve for the eigenvalues by the quadratic formula.

From now on, we shall assume the original matrix is of dimension greater than 2, and has been put into UUHM before the QR algorithm is applied.

Example 7.1

$$\text{Let } A = \begin{bmatrix} a & a & a & a \\ x & a & a & a \\ x & a & a & a \\ x & a & a & a \end{bmatrix}$$

We can choose U_1 as the ER such that $U_1 \begin{bmatrix} x \\ x \\ x \end{bmatrix} = \begin{bmatrix} -\sigma_1 \\ 0 \\ 0 \end{bmatrix}$, then form

$$\hat{U}_1 = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & U_1 \end{array}\right]$$

Premultiplying A by U_1 will change only the 2nd, 3rd, 4th rows of A. This gives

$$\hat{U}_1 A = \begin{bmatrix} a & a & a & a \\ -\sigma_1 & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \end{bmatrix} .$$

Post multiplying $\hat{U}_1 A$ by $\hat{U}_1^T (=\hat{U}_1)$ will change the 2nd, 3rd, 4th columns of $\hat{U}_1 A$ and we get

$$\hat{U}_1 A \ \hat{U}_1 = \begin{bmatrix} a & x & x & x \\ -\sigma_1 & x & x & x \\ 0 & \gamma & x & x \\ 0 & \gamma & x & x \end{bmatrix}$$

Now, we can choose U_2 as an ER (or PR) such that $U_2\begin{bmatrix}\gamma \\ \gamma\end{bmatrix} = \begin{bmatrix}-\sigma_2 \\ 0\end{bmatrix}$.

Let $U_2 = \begin{bmatrix} I_2 & 0 \\ 0 & U_2 \end{bmatrix}$. Again, if we premultiply and postmultiply $\hat{U}_1 A \hat{U}_1$ by $\hat{U}_2$, only the 3rd and 4th rows and columns are changed from $\hat{U}_1 A \hat{U}_1$, and we obtain $A_1 = \hat{U}_2 \hat{U}_1 A \hat{U}_1 \hat{U}_2$

$$= \begin{bmatrix} x & x & x & x \\ -\sigma_1 & x & x & x \\ 0 & -\sigma_2 & x & x \\ 0 & 0 & x & x \end{bmatrix} \varepsilon \text{UHM}$$

Example 7.2

$$\text{Suppose } A_1 = \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \end{bmatrix} \varepsilon \text{UUHM}$$

Apply QR algorithm to A_1. First, we need to form $A_1 = QR$, that is $Q^T A_1 = R$. We prefer to choose PR's P_{43}, P_{32}, P_{21} which wll eliminate a_{43}, a_{32}, a_{21} separately. Let $Q^T = P_{43}P_{32}P_{21}$, thus $Q^T A_1 = R$. Then postmultiply R by $Q = P_{21}^T P_{32}^T P_{43}^T$. This gives $A_2 = Q^T A_1 Q$

$$= P_{43}P_{32}P_{21}A_1P_{21}^TP_{32}^TP_{43}^T$$

$$= \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{bmatrix} P_{21}^TP_{32}^TP_{43}^T$$

$$= \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{bmatrix} P_{32}^TP_{43}^T$$

$$= \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & 0 & x \end{bmatrix} P_{43}^T$$

$$= \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \end{bmatrix} \varepsilon\text{UHM}$$

From this example, we observe that the QR algorithm preserves the Hessenberg form at each iteration. In addition, the number of multiplications and additions required to go from A_k to A_{k+1} is proportional to n^3 for a full matrix $A_1 \varepsilon R^{nxn}$, to n^2 for $A_1\varepsilon$UUHM. If A_1 is symmetric, the unreduced upper Hessenberg form of A_1 will be tridiagonal and the number of operations required is proportional to n.

CHAPTER VIII

THE IMPLICIT QR ALGORITHM

The next two algorithms are variants of algorithm 6.2. They are achieved without having to subtract the shifts from the diagonal and later restore them.

Theorem 8.1

Let A_1, $A,Q\varepsilon R^{nxn}$ with Q orthogonal and $A_1 \varepsilon UUHM$ having positive subdiagonal elements. If $A_1 = Q^T A\ Q$, then both A_1 and Q are uniquely determined by A and the first column of Q.

Proof. Let $A = [q_1, \ldots, q_n]$. We will prove this theorem by induction on k. Since

$$QA_1 = AQ \tag{8.1}$$

Then for k = 1 $q_1 a_{11} + q_2 a_{21} = A\ q_1$

$$q_1^T q_1 a_{11} + q_1^T q_2 a_{21} = q_1^T A\ q_1$$

$$q_2^T q_1 a_{11} + q_2^T q_2 a_{21} = q_2^T A\ q_1$$

where $q_2 = a_{21}^{-1}(A\ q_1 - q_1 a_{11})$ and $q_2^T q_2 = 1$.

We can solve the above equations to get

$$a_{11} = q_1^T A\ q_1 \quad \text{and then}$$

$$a_{21} q_2 = A' q_1 - q_1 a_{11} \tag{8.2}$$

In (8.2) q_1 determines q_2 up to a constant. The properties $q_2^T q_2 = 1$ and $a_{21} > 0$ then suffice to determine q_2 and a_{21} uniquely.

Assume we continue the above steps and have already computed $q_1, q_2, \ldots, q_k$, and the first k-1 columns of A_1. We want to compute the q_{k+1} and k^{th} column of A_1.

Formula (8.1) and $A_1 \varepsilon$UUHM imply

$$a_{k+1,k}q_{k+1}+a_{k,k}q_k + \ldots + a_{1k} \cdot q_1 = A\, q_k \qquad (8.3)$$

Multiply equation (8.3) by q_i^T to obtain $a_{ik} = q_i^T A\, q_k$ $(i = 1, \ldots, k)$ and $q_{k+1} = a_{k+1,k}^{-1}(A\, q_k - \sum_{i=1}^{k} a_{ik}q_i)$ with $q_{k+1}^T q_{k+1} = 1$. A_1 must have positive subdiagonal elements, so that q_{k+1} and $a_{k+1,k}$ can be uniquely determined. //

In fact, $a_{k+1,k}$ and q_{k+1} are determined up to a constant factor of absolute value unity simply by the requirement that $a_{k+1,k}$ be nonzero. It is this essential uniqueness of A_1 and Q that we shall actually use.

Theorem 8.1 tells us that we can determine a different Q_k at each step in algorithm 6.2 (call it $\hat{Q}_k$) but leave the first column of Q_k unchanged. By the uniqueness of the above theorem, we still can get the same Q_k and A_{k+1}.

In practice, the question is how to find $\hat{Q}_k$? Suppose we use one step of algorithm 6.2 with shift x_k to get $A_k - x_k I = Q_k R_k$. Let $Q_k = [q_1, q_2, \ldots, q_n]$.

Then $q_1 r_{11} = Q_k R_k e_1 = (A_k - x_k I) e_1 = a_1$, so $q_1 = r_{11}^{-1} a_1$, which is a multiple of a_1. Let P_k be a PR (or ER) such that $P_k a_1 = \|a_1\|_2 e_1$ where

$$a_1 = \begin{bmatrix} a_{11} - x_k \\ a_{21} \\ 0 \\ \vdots \\ 0 \end{bmatrix} .$$

Thus $P_k^T e_1 = a_1/\|a_1\|_2$. So that the first column of P_k^T is a multiple of a_1, and hence P_k^T has the same first column as Q_k.

By Theorem 7.1, we can always find orthogonal transformations $\hat{U}_{n-2}, \hat{U}_{n-1}, \ldots, \hat{U}_1$ such that $\hat{U}_{n-2}\hat{U}_{n-3} \ldots \hat{U}_1 P_k A_k P_k^T \hat{U}_1 \ldots \hat{U}_{n-3}\hat{U}_{n-2} \varepsilon \text{UHM}$.

Let $\hat{Q}_k = P_k^T \hat{U}_1 \ldots \hat{U}_{n-3}\hat{U}_{n-2}$, this is what we need. Observing the way we choose $\hat{U}_1, \ldots, \hat{U}_{n-2}$ in example 7.1, if we postmultiply P_k^T by $\hat{U}_1 \ldots \hat{U}_{n-2}$, the first column of P_k^T never can be disturbed, so that Q_k and $\hat{Q}_k$ have the same first column.

Algorithm 8.1

Choose an "origin shift" x_k. For k=1, 2, ..., calculate $\hat{Q}_k = P_k^T \hat{U}_1 \ldots \hat{U}_{n-2}$. Then form $A_{k+1} = \hat{Q}_k^T A_k \hat{Q}_k$.

Let us see some details of the practical computation by using this algorithm:

$$\text{Suppose} \qquad A_1 = \begin{bmatrix} a & a & a & a & a \\ a & a & a & a & a \\ 0 & a & a & a & a \\ 0 & 0 & a & a & a \\ 0 & 0 & 0 & a & a \end{bmatrix}$$

then $P_1A_1P_1^T$ has the form

$$\begin{bmatrix} a & a & a & a & a \\ a & a & a & a & a \\ x & a & a & a & a \\ 0 & 0 & a & a & a \\ 0 & 0 & 0 & a & a \end{bmatrix} \qquad (8.4)$$

where P_1 is the PR in (1,2)-plane. Hence $P_1A_1P_1^T$ is different from A_1 only on the first two rows and columns. Note that in (8.4) we have brought one nonzero element x in the (3,1) entry of A_1. Let $\hat{U}_1$ be the PR in the (2,3)-plane which makes x equal to zero. Then $\hat{U}_1P_1A_1P_1^T$ becomes

$$\begin{bmatrix} a & a & a & a & a \\ a & a & a & a & a \\ 0 & a & a & a & a \\ 0 & 0 & a & a & a \\ 0 & 0 & 0 & a & a \end{bmatrix} \qquad (8.5)$$

Postmultiply (8.5) by U_1, the above matrix will change only in the 2nd and 3rd columns and become

$$\begin{bmatrix} a & a & a & a & a \\ a & a & a & a & a \\ 0 & a & a & a & a \\ 0 & x & a & a & a \\ 0 & 0 & 0 & a & a \end{bmatrix} \qquad (8.6)$$

Again, we have gained a nonzero entry x in the (4.2) position of matrix (816). Thus we choose $\hat{U}_2$ as the PR in the (3,4)-plane to annihilate x in matrix (8.6). Then form $\hat{U}_2\hat{U}_1P_1A_1P_1^T\hat{U}_1\hat{U}_2$. Repeat the same step until we obtain another upper Hessenberg matrix. This means that we have finished one iteration.

One should note that it is convenient to take P_k, $\hat{U}_1$, ..., $\hat{U}_{n-2}$ as the PR's at each iteration because of the special form of A_k.

In algorithm 8.1, we do not compute A_{k+1} by using x_k explicitly. That is, we do not subtract x_k and then restore it. This method is not cheaper than shifting explicitly, since we have to compute $\hat{Q}_k$ at each iteration. One of the drawbacks of the algorithm 6.2 and 8.1 is that it may require complex origin shifts if A_1 is not symmetric. We will develop an implicit double shift technique from algorithm 8.1, which circumvents the problem of complex shifts by effecting two conjugate shifts simultaneously in real arithmetic. When A_1 is symmetric, the problem of complex shifts does not arise. In this case, the above two algorithms can be used. Since A_1 is a tridiagonal matrix, (we already assumed $A_1 \varepsilon$UUHM) this only requires a small number of operations.

The next algorithm is derived from the previous one, by applying two steps of algorithm 6.2 on A_{2k-1}($k = 1,2,\ldots$) with shifts x_{k1}, x_{k2} at each iteration. By theorem 6.1 we get $(A_{2k-1}-x_{k2}I)(A_{2k-1}-x_{k1}I) = Q_{k1}Q_{k2}R_{k2}R_{k1} = Q_kR_k$ and

$$A_{2k+1} = Q_k^T A_{2k-1} Q_k \qquad (8.7)$$

From the previous discussion, we need to determine an orthogonal matrix $\hat{Q}_k$ which has the same first column as Q_k. Form $\hat{Q}_k^T\hat{A}_{2k-1}\hat{Q}_k$, this gives the same Q_k and A_{2k+1} as in (8.7). As before, we know that the first column of Q_K is the same as that of $(A_{2k-1}-x_{k2}I)(A_{2k-1}-x_{k1}I)e_1$

$$= \begin{bmatrix} a_{11}-x_{k2} & a_{12} & \cdot \\ a_{21} & a_{22}-x_{k2} & \cdot \\ 0 & a_{32} & \cdot \\ \cdot & 0 & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} a_{11}-x_{k1} & a_{12} & \cdot \\ a_{21} & a_{22}-x_{k1} & \cdot \\ 0 & a_{32} & \cdot \\ \cdot & 0 & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} e_1$$

$$\begin{bmatrix} (a_{11}-x_{k1})(a_{11}-x_{k2})+a_{12}a_{21} \\ a_{21}(a_{11}-x_{k1})+a_{21}(a_{22}-x_{k2}) \\ a_{21}a_{32} \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} a_{11}^2-(x_{k1}+x_{k2})a_{11}+x_{k1}x_{k2}+a_{12}a_{21} \\ a_{21}(a_{11}-(x_{k1}+x_{k2})+a_{22}) \\ a_{21}a_{32} \\ 0 \end{bmatrix} \qquad (8.8)$$

It is convenient to take these two shifts as the eigenvalues of the matrix

$$\begin{bmatrix} a_{n-1,n-1} & a_{n-1} \\ a_{n,n-1} & a_{n,n} \end{bmatrix}$$

at each iteration, we have $x_{k1}+x_{k2} = a_{n-1,n-1}+a_{n,n}$ and $x_{k1} \cdot x_{k2} = a_{n-1,n-1} \cdot a_{n,n} - a_{n-1,n} \cdot a_{n,n-1}$. Substitute these into (8.8). We obtain

$$a_1 = \begin{bmatrix} \dfrac{a_{21}(a_{11}^2-a_{n-1,n-1}a_{11}-a_{nn}a_{11}+a_{n-1,n-1}a_{nn}-a_{n-1,n}a_{n,n-1})}{a_{21}+a_{12}} \\ \\ a_{21}(a_{11}+a_{22}-a_{n-1,n-1}-a_{nn}) \\ a_{21}a_{32} \\ 0 \\ . \\ . \end{bmatrix}$$

There are three nonzero elements in a_1 and those involve only real arithmetic no matter whether the shifts are complex numbers or not.

The remainder of the steps are the same as that in algorithm 8.1, except that now we must use ER's: Let U_k be the ER such that $U_k a_1 = \|a_1\|_2 e_1$. Then $U_k^T e_1 = a_1/\|a_1\|_2$. Thus U_k^T will have the same first column as Q_k. Form $U_k A_{2k-1} U_k^T$ and we can find orthogonal transformations $\hat{U}_{n-2}, \hat{U}_{n-1}, \ldots, \hat{U}_1$ such that $\hat{U}_{n-2}\hat{U}_{n-3}, \ldots, \hat{U}_1 U_k A_{2k-1} U_k^T \hat{U}_1 \ldots \hat{U}_{n-3}\hat{U}_{n-2} \varepsilon \mathrm{UHM}$. If we choose $\hat{Q}_k = U_k^T \hat{U}_1 \ldots \hat{U}_{n-2}$, then $\hat{Q}_k$ will have the same first column as that of Q_k. Now we summarize the above procedure as the following:

Algorithm 8.2

Implicitly doubly shifted QR algorithm. Choose two "origin shifts" x_{k1}, x_{k2}. For $k = 1, 2, \ldots$, determine $\hat{Q}_k$, then form $A_{2k+1} = \hat{Q}_k^T A_{2k-1} \hat{Q}_k$.

Let us see how to carry out one iteration of this algorithm in practice:

$$A_1 = \begin{bmatrix} a & a & a & a & a \\ a & a & a & a & a \\ 0 & a & a & a & a \\ 0 & 0 & a & a & a \\ 0 & 0 & 0 & a & a \end{bmatrix} \quad \text{then}$$

$$U_1 A_1 U_1^T = \begin{bmatrix} a & a & a & a & a \\ a & a & a & a & a \\ x & a & a & a & a \\ x & x & a & a & a \\ 0 & 0 & 0 & a & a \end{bmatrix} \qquad (8.9)$$

Choose $\hat{U}_1$ as an ER such that it annihilates the (3.1) and (4.1) positions in (8.9). $\hat{U}_1$ has the form:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & u & u & u & 0 \\ 0 & u & u & u & 0 \\ 0 & u & u & u & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Form $\hat{U}_1 U_1 A_1 U_1^T$ will change the 2nd, 3rd and 4th rows of the matrix in (8.9) and give

$$\begin{bmatrix} a & a & a & a & a \\ a & a & a & a & a \\ 0 & a & a & a & a \\ 0 & x & a & a & a \\ 0 & 0 & 0 & a & a \end{bmatrix} \qquad (8.10)$$

Postmultiplying matrix in (8.10) by $\hat{U}_1$ will change the 2nd, 3rd and 4th columns of it and leave the first column unchanged, thus $\hat{U}_1 U_1 A_1 U_1^T \hat{U}_1$ is

$$\begin{bmatrix} a & a & a & a & a \\ a & a & a & a & a \\ 0 & a & a & a & a \\ 0 & x & a & a & a \\ 0 & x & x & a & a \end{bmatrix} \qquad (8.11)$$

Compare (8.9) and (8.11) in both matrices, we have obtained three nonzero elements in the original Hessenberg matrix, but in different positions. Now we can choose $\hat{U}_2$ as an ER which annihilates the (4,2) and (5,2) entries in (8.11) which has the form

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & u & u & u \\ 0 & 0 & u & u & u \\ 0 & 0 & u & u & u \end{bmatrix}$$

Then compute $\hat{U}_2\hat{U}_1U_1A_1U_1^T\hat{U}_1\hat{U}_2$, this gives

$$\begin{vmatrix} a & a & a & a & a \\ a & a & a & a & a \\ 0 & a & a & a & a \\ 0 & 0 & a & a & a \\ 0 & 0 & x & a & a \end{vmatrix}$$

For the last step, we can choose a PR instead of an ER since we only need to make (5,3) entry equal to zero. We choose P_{45} and form $P_{45}\hat{U}_2\hat{U}_1U_1A_1U_1^T\hat{U}_1\hat{U}_2P_{45}$ UHM, thus we have finished one iteration of this algorithm on A_1.

In using this algorithm, we paid for calculating a_1 on each step. In addition, the multiplication of reducing $U_kA_{2k-1}U_k^T$ to $A_{2k+1}\epsilon$UHM is $\sim n^2$. If we can use real numbers as the shifts, and we shift twice separately by using algorithm 6.2, the cost from A_{2k-1} to $A_{2k+1}\epsilon$UHM is also $\sim n^2$. Of course, nothing is saved by this technique if A_1 is symmetric.

Parlett [2] has shown that if the shifts are not the eigenvalues and if A_{2k-1} has nonzero subdiagonal elements, then so does A_{2k+1}. This tells us that the recursion will not end prematurely. In fact, when we do the practical computation, we will get a "sufficiently small" subdiagonal element, say $a_{j+1,j}$. If $j = n-1$ or $n-2$, then we have obtained one or two eigenvalues. Thus our program will do the deflation and apply the algorithm to the matrix in $R^{(n-1)x(n-1)}$ or $R^{(n-2)x(n-2)}$ automatically. If $j \leqslant n-3$

the program should continue the algorithm on the lowest submatrix.

CHAPTER IX

SINGULAR AND INVARIANT MATRICES

Singular Matrices

If $A_1 \varepsilon UUHM$ and is singular, then the QR decomposition of A_1 is not uniquely determined. However, we still can apply the QR algorithm to it as we will see in the following:

Since A_1 = QR, then

$$a_k = \sum_{k=1}^{i} q_k r_{ki} \quad i = 1, \ldots, n-1, \text{ and}$$

$$|r_{ii}| = \|a_i - \sum_{k=1}^{i-1} q_k r_{ki}\|_2 / \|q_i\|_2$$

$$= \|a_i - \sum_{k=1}^{i-1} q_k r_{ki}\|_2$$

Since $A_1 \varepsilon UUHM$ and the first n-1 columns of A_1 are linearly independent, therefore $r_{ii} \neq 0$ i=1, ..., n-1. This gives us that $r_{nn}=0$ ($\because$ A_1 is singular).

To get A_2 = RQ we have $a_{n,n-1} = r_{n,n} \; q_{n,n-1} = 0$ and $a_{n,n} = r_{n,n} q_{n,n} = 0$. This tells us that after one transformation, a zero eigenvalue is revealed if A_1 is singular. Thus the singular matrix case is a very fortunate case.

If we do the shifted algorithms on A, then $A_1 = A-x_1 I$ is singular if the shift x_1 is an eigenvalue of A. In fact, for both explicitly and implicitly shifted algorithms, we also can obtain an eigenvalue x_1 after one iteration. This means we made a very good guess.

Invariant Matrices

In this section, we will describe all matrices which are invariant under the QR algorithm. First, we just state without proof some definitions and theorems from linear algebra which are important prerequisites for the upcoming material. The interested reader can refer to [4] for the details.

Definition 9.1

A matrix A is nonderogatory if the minimal polynomial equals the characteristic polynomial. Otherwise, A is said to be derogatory.

Theorem 9.1

Every matrix A (over the complex numbers) is similar to a matrix J in Jordan canonical form.

Theorem 9.2

A matrix A is nonderogatory if the distinct Jordan blocks of its Jordan canonical form correspond to distinct eigenvalues.

Theorem 9.3

An nxn matrix A is nonderogatory if for every complex number λ, the matrix $A-\lambda I$ has rank at least n-1.

Theorem 9.4 (Cayley-Hamilton)

Let P be the characteristic polynomial of A, then $P(A) = 0$

$$\text{Let} \quad J = \begin{bmatrix} J_1 & & & 0 \\ & J_2 & & \\ & & \ddots & \\ 0 & & & J_r \end{bmatrix}_{nxn} \qquad Q' = \begin{bmatrix} Q_{1I} & \cdots & Q_{1r} \\ \vdots & Q_{22} \ddots & \vdots \\ Q_{ri} & \cdots & Q_{rr} \end{bmatrix}_{nxn} \qquad (*)$$

where J_i (i=1, ..., r) are the Jordan blocks of J and $\lambda_i \neq \lambda_j$

Lemma 1

If $JQ' = Q'J$, then Q' is block diagonal; that is,

$$Q' = \begin{bmatrix} Q_{11} & & & 0 \\ & Q_{22} & & \\ & & \ddots & \\ 0 & & & Q_{rr} \end{bmatrix} .$$

Lemma 2

If $J_iQ_{ii} = Q_{ii}J_i$ then Q_{ii} is upper triangular, and Q_{ii} has the form

$$\begin{bmatrix} a_i & b_i & \cdots & m_i \\ & \ddots & \ddots & \vdots \\ & & \ddots & b_i \\ 0 & & & a_i \end{bmatrix}$$

Theorem 9.5

Q' and J^n have the same block diagonal form if $JQ' = Q'J$.

The main theorem of this section is the following:

Theorem 9.6

Let $A_1 \varepsilon UUHM$, and $A_1 \varepsilon R^{nxn}$, then A_1 is invariant under the unshifted QR algorithm if A_1 is a scalar multiple of an orthogonal matrix.

Proof (→). As $A_1 \varepsilon UUHM$, then A_1 is nonderogatory. Suppose A_1 is invariant under the unshifted QR algorithm, that is $A_1 = QR$ and $A_2 = RQ = A_1$. Thus

$$QA_1 = A_1 Q \qquad (9.1)$$

Let P be the characteristic polynomial of A_1, and let J be the Jordan canonical form of A_1 (there exists a nonsingular matrix S such that $S^{-1}A_1 S = J$). By the Cayley-Hamilton theorem, $P(A_1) = P(J) = 0$, where $P(J) = a_n J^n + \ldots + a_1 J + a_o I = 0$. Then

$$-a_n J^n = a_{n-1} J^{n-1} + \ldots + a_1 J + a_o I \qquad (9.2)$$

Since $A_1 = SJS^{-1}$, we can change (9.1) into

$$JQ' = Q'J \qquad (9.3)$$

where $Q' = S^{-1}QS$.

It follows from theorem 9.5 and formula (9.2) that $A' = a'_{n-1}J^{n-1}+a'_{n-2}J^{n-2}+ \dots +a'_1J+a'_oI$. We can change Q' back to Q to get

$$Q = a'_{n-1}A_1^{n-1}+ \dots +a'_1\ A_1+a'_oI\ .$$

This tells us that Q is a polynomial in A_1 of degree less than or equal to n-1. But $Q = AR^{-1}$ implies that $Q\varepsilon UUHM$, hence we have

$$Q = a'_1A_1+a'_oI \qquad (9.4)$$

Since $A_1 = QR = (a''_1A_1+a'_oI)\cdot R$, this gives $r_{11}^{-1}a_{11} = a'_1a_{11}+a'_o$ and $r_{11}^{-1}a_{21} = a'_1a_{21}$. Solve for a'_1 and a'_o, we have $a'_o = 0$ and $a'_1 = r_{11}^{-1}$. Therefore formula (9.4) becomes $Q = r_{11}^{-1}A_1$, that is $A_1 = r_{11}Q$.

($\leftarrow$) If $\alpha^{-1}A_1$ is orthogonal, then let $Q = \alpha^{-1}A_1$. Thus $A_1 = \alpha Q$ and then $R = \alpha I$, this implies A_1 is invariant.

//

For the shifted QR algorithm, we have the similar result (see [5]):

"A_1 is invariant under the doubly shifted QR algorithms if $A_1^2-\sigma A_1+\rho I$ is a multiple of an orthogonal matrix. Where σ and ρ are parameters depending only on the choice of origin shifts."

If a matrix is invariant under the QR algorithm, some special treatments must be used (see [3]).

CHAPTER X

AN APPLICATION OF THE QR ALGORITHM

Stiff Differential Equations

For any single differential equation

$$y' = \lambda y+b \tag{10.1.1}$$

we say this is a stiff differential equation [6], if $Re(\lambda)<0$ and large in absolute value. We can solve (10.1.1) explicitly, and obtain the exact solution $y = ce^{\lambda t}+\phi(t)$, where $\phi(t)$ is a particular solution to the nonhomogeneous case. It can be shown that the exact solution will have a constant term and another term which decays rapidly if the differential equation is stiff. Therefore, if we solve a stiff differential equation, we may expect the numerical solutions approach the constant solution after a very short time period. So that we can use special methods which allow us to take a bigger stepsize in the computer program, in order to minimize the numerical work and save the computing time.

For the nonlinear single differential equation

$$y' = f(y) \tag{10.1.2}$$

Suppose f is twice continuously differentiable, by a Taylor series expansion about y_o, we obtain $y' = f(y_o)+f'(y_1)(y-y_o)+f''(\xi)(y-y_o)^2/2$. If y is close to y_o, then

$$y' = \lambda y+b \qquad (10.1.3)$$

where $\lambda = f'(y_o)$ and $b = f(y_o)-f'(y_o)y_o$.

By the well-posedness property, the numerical solutions of (10.1.3) are close to those in (10.1.2). Hence we can consider the linearized case instead of the original nonlinear case, and have the same discussion as before.

Determining Stiffness of a System of Differential Equations

In general, we want to extend the previous discussion to a system of differential equations. Suppose we have a system of n linear differential equations, it is convenient to write the system in the matrix form:

$$Y' = AY+B \qquad (10.2.1)$$

If we use numerical methods for solving this system, we would like to know whether or not this is a stiff system, so that we will know whether we must use some special methods which have been developed for stiff systems. In the following, we will see that this can be done by examining the eigenvalues of matrix A:

Assume A is simple (otherwise, we can perturb it slightly to make it simple), then there exists a nonsingular matrix T and a diagonal matrix D such that $A = TDT^{-1}$, where $D = \text{diagonal}[\lambda_1, \lambda_2, \ldots, \lambda_n]$. Now, we can change the variables in (10.2.1) by multiplying both sides on the equation by T^{-1} to get $T^{-1}Y' = T^{-1}AY+T^{-1}B$. Equation (10.2.1) would become $\hat{Y}' = D\hat{Y}+\hat{B}$ where $\hat{Y}' = T^{-1}Y'$ and $\hat{B} = T^{-1}B$. This can also be written as:

$$\hat{y}_i' = \lambda_i \hat{y}_i + \hat{b}_i \qquad i = 1, \ldots, n \qquad (10.2.2)$$

Hence the new refined form (10.2.2) has been reduced to n separate single linear differential equations, and each λ_i is an eigenvalue of A. If there exists one or more eigenvalues of A having negative real parts and some of them are relatively large in absolute values, then (10.2.2) is a stiff system.

For the system of nonlinear differential equations

$$y_i' = f_i(x, y_1, y_2, \ldots, y_n) \qquad i = 1, \ldots, n \quad (10.2.3)$$

suppose f_i is twice continuously differentiable. We can linearize (10.2.3) at $x_o, y_o)$ and get

$$y_i' = \sum_{j=1}^{n} \frac{\partial f_i}{\partial y_j} y_j + b_i \qquad i = 1, 2, \ldots, n$$

where $b_i = f_i(x_o, y_o) + \frac{\partial f_i}{\partial x} \cdot (x - x_o) - \sum_{j=1}^{n} \frac{\partial f_i}{\partial y_j} \cdot y_{jo}$.

Changing the linearized system into the matrix form, we obtain Y' = JY+B and J is the Jacobian matrix of f_i's.

That is

$$J = \begin{vmatrix} \frac{\partial f_1}{\partial y_1} & \cdots\cdots\cdots & \frac{\partial f_1}{\partial y_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial y_1} & \cdots\cdots\cdots & \frac{\partial f_n}{\partial y_n} \end{vmatrix}_{n x n}$$

We can use difference equations to get the derivatives in J by using

$$\frac{\partial f_i}{\partial y_j} \approx [f_i(x, y_1, \ldots, y_j+\delta, \ldots, y_n) - f_i(x, y_1, \ldots$$

$$y_j-\delta, \ldots, y_n)]/2\delta \quad .$$

Now we can apply the QR algorithm on J, and examine the eigenvalues of J.

BIBLIOGRAPHY

1. G. W. Stewart. Introduction to matrix computation. Academic Press, 1973.

2. B. N. Parlett. "The LU and QR algorithms." In A. Ralston and H. S. Wilf (Eds.). Mathematical methods for digital computers. Vol. 2. Wiley, 1967.

3. J. H. Wilkinson. The algebraic eigenvalue problem. Oxford University Press, 1965.

4. F. R. Gantmacher. The theory of matrices. Chelsea, 1960.

5. B N. Parlett. "Singular and invariant matrices under the QR transformation." Math. Comput., 1966.

6. L. F. Shampine and C. W. Gear. "A user's view of solving stiff ordinary differential equations." SIAM Review 21:1-17, 1979.