The QR Algorithm

Marco Latini

February 26, 2004

Contents

1 Introduction

In our presentation of the QR algorithm we follow closely the presentation found in Stewart's [20] textbook on Matrix algorithms.

All eigenvalue algorithms are necessarily iteratively. This follows from Abel's famous proof that there is no algebraic formula for the roots of a general polynomial of degree greater than four. Specifically to any polynomial $p(x) = x^n \sum_i a_i x^i$ there exist a companion matrix C_p which is as follows

$$C_p = \begin{bmatrix} a_{n-1} & a_{n-2} & \dots & a_1 & a_0 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}.$$

Notice that the characteristic polynomial of this matrix is $p(x)$ and therefore if an algorithm existed to determine the eigenvalues in a finite number of steps we would have in effect found a formula for the roots of the polynomial problem.

Therefore the most used algorithms for the eigenvalue problems are iterative and they are based on powers of a matrix. The most famous of the algorithms is the QR algorithms. This algorithms can produce a Schur form from a general matrix in $O(n^3)$ steps. The algorithm can be tuned to compute eigenvalues of special symmetric or tri-diagonal matrices and the algorithms itself always suggests new methods and new theories.

The QR algorithm is based on two algorithms, the power method and the inverse power method. The QR comes in many different shapes and forms and we will introduce our "base" case as the explicitly shifted QR algorithm.

2 The Power and Inverse Power method

2.1 The Power Method

Let A be a non-defective matrix and let (λ_i, x_i) be the eigen-value eigen-vector pair. Notice that the set of eigenvectors forms a linearly independent set that spans the entire space. Thus any vector u can be expressed as a linear combination of the eigenvectors.

Now let $|\lambda_1| > |\lambda_i|$, then $A^k u$ results in a product of eigenvectors eigenvalues

$$A^k u = \gamma_1 \lambda_1^k x_1 + \sum_i \gamma_i \lambda_i^k x_i. \tag{1}$$

As $k \to \infty$ the term containing λ_1 will dominate and $A^k u$ approaches a multiple of the dominant eigenvector.

Based on these ideas we can write the power iteration.

Choose $v^{(0)}$ such that $||v^{(0)}|| = 1$, then for $k = 1, 2, ...$ we do the following steps

$$\begin{aligned} w &= Av^{(k-1)} \\ v^{(k)} &= \frac{w}{||w||} \\ \lambda^{(k)} &= (v^{(k)})^T A v^{(k)}. \end{aligned}$$

The rate of convergence to the eigenvectors

$$
\begin{aligned}
||v^{(k)} - q_1|| &= O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right) \\
|\lambda^{(k)} - \lambda_1| &= O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right).
\end{aligned}
$$

The power iteration is of limited use since it can only find the eigenvector corresponding to the dominant eigenvalue and the convergence of the eigenvector is only linear. Furthermore the quality of this convergence deteriorates if $\lambda_1 \approx \lambda_2$. The advantage of the power method is that it requires only matrix-vector multiplications and it is globally convergent provided the matrix has a dominant eigenvalue.

The power method goes back to a work by Muntz of 1913 [9]. The method was then used by Hotelling in 1933 [3, 4] and by Aitken in 1937 [7]. Before the QR era the power method was the standard tool to compute eigenvalues.

2.2 The Inverse power method

To overcome some of the short-comings of the power iteration the inverse power method is introduced. Technically the inverse power iteration can be described as a "shift-and-invert" method. This method is based on the following insight. Suppose μ is NOT an eigenvalue of A. Then the matrix $B = (A - \mu I)^{-1}$ has the same eigenvectors as A and the eigenvalues are $(\lambda_j - \mu)^{-1}$ where λ_j is an eigenvalue of A. Thus if we choose a value of μ close to an eigenvalue then $(\lambda_j - \mu)^{-1}$ will be an enormous number. If we thus apply the power iteration to the matrix $(A - \mu I)^{-1}$ we will converge rapidly to q_j and to $(\lambda_j - \mu)^{-1}$.

Thus we present the inverse iteration. Choose $v^{(0)}$ such that $||v^{(0)}|| = 1$ and for $k = 1, 2, ...$ we have

$$
\begin{aligned}
w &= (A - \mu I)^{-1} v^{(k-1)} \\
v^{(k)} &= \frac{w}{||w||} \\
\lambda^{(k)} &= (v^{(k)})^T A v^{(k)}.
\end{aligned}
$$

The convergence for this method are as follows

$$
\begin{aligned}
\left|\left|v^{(k)} - q_j\right|\right| &= O\left(\left|\frac{\mu - \lambda_j}{\mu - \lambda_k}\right|^k\right) \\
\left|\lambda^{(k)} - \lambda_j\right| &= O\left(\left|\frac{\mu - \lambda_j}{\mu - \lambda_k}\right|^{2k}\right),
\end{aligned}
$$

where λ_j is the closest eigenvalue to μ and λ_k is the second closest.

The inverse power method is due to Wielandt in 1944 [22].

2.3 The Rayleigh quotient iteration

Notice that the power method can be used most effectively to compute the dominant eigenvalue while the inverse power method can best be used to compute eigenvectors. If we combine the two we obtain the Rayleigh quotient iteration.

The main insight in this method is furnished by the fact that we can use the estimates from the inverse power iteration $\lambda^{(k)}$ directly as shifts for the inverse power method itself!

Choose $v^{(0)}$ as any vector such that $||v^{(0)}|| = 0$. Let $\lambda^{(0)} = (v^{(0)})^T A v^{(0)}$, then for $k = 1, 2, ...$ we have

$$\begin{aligned} w &= (A - \lambda^{(k-1)} I)^{-1} v^{(k-1)} \\ v^{(k)} &= \frac{w}{||w||} \\ \lambda^{(k)} &= (v^{(k)})^T A v^{(k)}. \end{aligned}$$

The convergence for this method is given by

$$\begin{aligned} ||v^{(k+1)} - q_j|| &= O\left(\left|\left| v^{(k)} - q_j \right|\right|^3 \right) \\ \left| \lambda^{(k+1)} - \lambda_j \right| &= O\left(\left| \lambda^{(k)} - \lambda_j \right|^3 \right). \end{aligned}$$

Third order convergence is extremely fast.

The Rayleigh quotient iteration is due to Ostrowski in 1958, 1959 [11, 12, 13, 14, 15, 16].

3 The explicit QR algorithm with shifts

The explicit QR algorithm with shifts can be written as follows for any matrix A.

Let $A_0 = A$, and let $k = 1, 2, 3, ...$ then given shifts κ_i

$$\begin{aligned} A_k - \kappa_k I &= Q_k R_k \\ A_{k+1} &= R_k Q_k + \kappa_k I. \end{aligned}$$

Notice that the iterates satisfy

$$\begin{aligned} A_{k+1} &= R_k Q_k + \kappa_k I \\ &= Q_k^* (A_k - \kappa_k I) Q_k + \kappa_k I \\ &= Q_k^* A Q_k, \end{aligned} \tag{2}$$

and therefore are related through a similarity transformation.

The algorithm works because of its connection to the inverse power method with shift κ_k.

3.1 The QR algorithm and the Inverse Power Method

The main idea is the deflation of a Schur decomposition in the presence of an eigenvalue, eigenvector pair (λ, q). Let $Q = (Q_p q)$ be unitary. Then

$$Q^* A Q = \begin{pmatrix} Q_p^* A Q_p & Q_p^* A q \\ q^* A Q_p & q^* A q \end{pmatrix} = \begin{pmatrix} B & h \\ g & \mu \end{pmatrix},$$

where $g = q^* A Q_p = \lambda q^* Q_p = 0$ and $\mu = q^* A q = \lambda q^* q = \lambda$.

Now we apply the same reasoning to the QR Algorithm. Consider a step of QR. Consider the following decomposition of $Q(A - \kappa I) = R$ with

$$\begin{pmatrix} Q_p^* \\ q^* \end{pmatrix} (A - \kappa I) = \begin{pmatrix} R_p \\ r^* \end{pmatrix}. \tag{3}$$

Notice that $r^* = r_{nn} e_n^*$ and $q^* = r_{nn} e_n^* (A - \kappa I)^{-1}$. Thus the last column of the matrix Q (which we call q^*) is the result of the inverse power method with shift κ applied to the vector e_n^*. This analysis suggests the following decomposition of A

$$A = \begin{pmatrix} B & h \\ g^* & \mu \end{pmatrix}. \tag{4}$$

Therefore notice that when we apply one iteration of QR is like applying a step of Schur factorization with the eigenvalue, eigenvector pair (μ, e_n). Therefore $\mu = e_n^* A e_n$ is a good candidate for a shift and in fact this is called the Rayleigh quotient shift.

In order to show that the QR Algorithm converges we need to show that as we iterate with the above shift, $g \to 0$ and $\mu \to \lambda$.

Consider the following partition

$$A - \kappa I = \begin{pmatrix} B - \kappa I & h \\ g^* & \mu - \kappa \end{pmatrix} = QR = \begin{pmatrix} P & f \\ e^* & \pi \end{pmatrix} \begin{pmatrix} S & r \\ 0 & \rho \end{pmatrix}. \tag{5}$$

Similarly we write the next step of the iteration as follows

$$RQ = \begin{pmatrix} S & r \\ 0 & \rho \end{pmatrix} \begin{pmatrix} P & f \\ e^* & \pi \end{pmatrix} = \hat{A} - \kappa I = \begin{pmatrix} \hat{B} - \kappa I & \hat{h} \\ \hat{g}^* & \hat{\mu} - \kappa \end{pmatrix}. \tag{6}$$

Notice that from $||e||^2 + \pi^2 = ||f||^2 + \pi^2 = 1$ we have that $||e|| = ||f||$. Next we want to show that e is small when g is small. We get that $g^* = e^* S$ and therefore $||e|| \leq ||S^{-1}||||g||$. We let $\sigma = ||S^{-1}||$. We now need a bound on $\rho = f^* h + \pi(\mu - \kappa)$. We have that $||\rho|| \leq \sigma ||g||||h|| + |\mu - k|$.

Using the result for ρ and for e we can find a bound on g to show that

$$||\hat{g}|| = ||\rho e^*|| \leq \sigma^2 ||h||||g|| + \sigma(\mu - \kappa)||g||. \tag{7}$$

With this we can show that as the iteration proceeds choosing μ close enough to λ and g_0 sufficiently small then $\hat{g} \to 0$ and $\mu \to \lambda$. This is a local proof of the convergence of the QR algorithm. There are no global proofs of convergence for the QR algorithm. Once one eigenvalue is found in principal we can deflate the matrix and re-apply the QR algorithm to the sub-matrix.

How fast is the convergence? We can show that under the bounding $h_k \leq \eta$ and $\sigma_k \leq \sigma$ then

$$||\hat{g}|| \leq \sigma^2 \eta ||g||^2, \tag{8}$$

therefore the convergence is quadratic. For A_0 hermitian, then $h_k = g_k$ and we have that

$$||\hat{g}|| \leq \sigma^2 ||g||^3. \tag{9}$$

3.2 The unshifted QR algorithm and the power iteration

As the QR algorithm is applied it provides an excellent approximation to the eigenvector on the bottom right hand corner. However, it is observed that excellent approximations are also obtained for the other elements on the main diagonal. Therefore it appears that if we apply the QR algorithm ad infinitum all the elements on the diagonal will indeed become eigenvalues although the convergence is very slow. This fact can be explained if we consider the connection between the QR algorithm and the power iteration.

This is a manifestation of a deeper theorem that applies to the unshifted QR algorithm and shows that the un-shifted QR algorithm can provide a Schur factorization of a matrix A. More technically, suppose $A = X\Lambda X^{-1}$ where the matrix possess a complete set of distinct eigenvalues. Let $X^{-1} = LU$ and $X = QR$. If A_k, the kth step in the QR iteration has decomposition $A_k = Q_k R_k$ then there exist diagonal matrices D_k where $|D_k| = I$ such that $Q_k D_k \to Q$. This theorem is due to Wilkinson and dates back to 1965 [25, 24].

Therefore the above theorem proves that the unshifted QR algorithm produces a sequence of matrices tending to a triangular matrix.

Notice that it is not always the case the $X^{-1} = LU$. When this is not the case, the algorithm converges in finite arithmetic but the convergence can be very slow. This phenomenon is referred to as disorder in the eigenvalues because it corresponds to the eigenvalues that are not presented in descending order.

The above results can be used to explain the convergence observed in the shifted QR algorithm. In particular notice that the shifted QR algorithm is like applying the un-shifted QR algorithm to the matrix $A - \kappa_k I$. This explain why the elements in the sub-diagonal of A are reduced as $\mu \to \lambda$ and as the iteration proceeds.

3.3 Origin of the QR algorithm

The QR algorithm dates back to the Rutishauser in 1955 [17] was the first to propose what is now known as the LU algorithm to determine the eigenvalues of a matrix. He proposed that for a tridiagonal matrix T one could do an LU factorization and multiply back in reverse order. In 1958 Rutishauser [18] introduced a shift to speed the convergence. However, this algorithm is unstable if the matrices are not positive definite.

In 1961 Kublanovskaya [6] and Francis [1, 2] independently proposed to substitute the stable QR decomposition instead of the LU decomposition. Francis went on to propose the shift and the reduction to Hessenberg form.

The connection between the QR algorithm and the inverse power iteration is first mentioned in Stewart 1973 [19]. The connection with the power method was known right from the beginning.

The overall stability theory for the QR algorithm was given by Wilkinson in 1965 in his work "the algebraic eigenvalue problem" [25, 24]

4 The Hessenberg form

It is now to focus on the cost of the QR algorithm. Notice that the algorithm requires a decomposition $A = QR$ which takes $O(n^3)$ operations. Because this needs to be repeated for each eigenvalue we obtain an overall cost of $O(n^4)$ which is prohibitively expensive.

The cure to this problem is the transformation of the matrix to upper-Hessenberg form H. When this is done the total cost of the algorithm is $O(n^3)$.

4.1 Householder transformation

To reduce a matrix to Hessenberg form Householder reflections are used. These reflections were introduced by Householder in 1958 [5]. However, their applications to eigenvalue problems is due to Wilkinson 1960 [23].

We defined a Householder reflection as

$$H = I - uu*, \tag{10}$$

where $||u|| = \sqrt{2}$. Notice that a householder matrix is Hermitian and unitary. The main use of householder reflections is to introduce zeros into vectors (or matrices). Consider the following application. Choose $u = (a + e_1)/\sqrt{1 + a_1}$ for any vector a such that $||a|| = 1$. Then when we apply the Householder reflection we obtain $Ha = -e_1$.

We can thus reduce the matrix A into upper Hessenberg form via Householder reflections applied to the matrix A as follows. Let

$$A = \begin{pmatrix} \alpha_{11} & a_{12} \\ a_{21} & A_{22} \end{pmatrix},$$

choose the householder reflection H_1 such that $H_1 a_{21} = \nu_1 e_1$ then

$$H_1 A H_1 = \begin{pmatrix} \alpha_{11} & a_{12} H_1 \\ \nu e_1 & H_1 A_{22} H_1 \end{pmatrix},$$

thus we have annihilated the elements below the first column via a similarity transformation. We can repeat this process for all steps to obtain a upper Hessenberg form. The cost of the operation is $O(n^3)$.

Now let $Q = H_1 H_2 ... H_{n-2}$ then the first column of Q is e_1.

4.2 Givens rotations

The Givens rotations are another set of orthogonal transformations that can be used to reduce a matrix into upper Hessenberg form. In particular the Givens rotations are used to produce the QR factorization of the upper Hessenberg matrix H. They are cheaper to apply than Householder transformations when we apply a step of the QR algorithm.

Given rotations are defined as follows

$$P = \begin{pmatrix} c & s \\ -\bar{s} & \bar{c} \end{pmatrix} \tag{11}$$

where c and s are the cosine and sine of a rotation by θ degrees. Givens rotations are used to introduce a zero in a two component vector such that

$$P \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix}, \tag{12}$$

where $g = \nu a/|a|$, $\nu = \sqrt{a^2 + b^2}$ and $c = |a|/|\nu|$ and $s = a\bar{b}/(|a|\nu)$. This definition of c is guaranteed to leave the cosine real. We can generalize this procedure and apply the Givens rotations in larger matrices to selectively introduce a zero in appropriate locations.

Notice that the QR factorization via Givens rotations takes $O(n^2)$ operations.

Notice that givens rotations are also used to move a non-zero element around in a matrix by successively annihilating it in one position and letting it pop up in another position. This process is called chasing the bulge.

4.3 Invariance of the Hessenberg form in QR Algorithm

The Hessenberg transformation is chosen because after one sweep of the QR algorithm the matrix remains in upper Hessenberg form.

Notice that the QR step is given by reduction to triangular via orthogonal transformations $H = QR$ where $Q = P_{12}P_{23}...$ and the P_{12} are the Givens rotations. The second step in the algorithm is $\hat{H} = RQ$ and therefore we are right-multiplying by the Givens rotations again. This process preserves the Hessenberg structure so that $\hat{H}$ is an upper Hessenberg matrix. This was first noticed by Francis in 1961 [1, 2].

5 Some Techniques to accelerate convergence in the QR algorithm

5.1 Deflation

There are many ways convergence can be accelerated in QR. One way is to use deflation, once we detect a negligible sub-diagonal. Notice that if we do this trick we still need to apply the transformation to the entire overall matrix if we wish to find a Schur decomposition of the matrix A. If we only wish to find the eigenvalues of the matrix then we can completely forget about the rest of the matrix and use the problem.

5.2 The Wilkinson shift

As mentioned earlier the natural choice for the shift is the Rayleigh shift defined as the a_{nn} element of the matrix. However, the Rayleigh quotient is real and cannot approximate complex eigenvalues. For this reason we choose the Wilkinson shift which is defined as the eigenvalue of the lower 2×2 matrix that is closest to the a_{nn} value. This shift can move the entire QR iteration in the complex plane. This method requires the solution of a 2×2 eigenvalue problem but this can be done directly using the algebraic formula.

This shift is due to a work by Wilkinson in 1968 [26].

5.3 Obtaining the eigenvectors

Notice that the QR algorithm yields the Schur factorization $A = QTQ^*$ of a matrix. The eigenvectors are found by locating the eigenvectors of the triangular matrix T which we call X and then multiplying by Q to obtain QX the eigenvectors of A.

Notice that sometimes in computations we may run into the problem that eigenvalues vary quite a bit in size. But because eigenvectors are determined up to a scalar multiple we can re-size them.

5.4 Ad hoc shifts

Notice that convergence of the QR algorithm with shifts is not guaranteed. It may happen that for a certain matrix the algorithm cannot converge. When this happen the idea is to restart the iteration with a new shift called the "ad hoc" shift. This shifts works many time and to this day it is not completely understood why it works.

5.5 Aggressive deflation

Aggressive deflation is a technique used when we encounter in the sub-diagonal two consecutive elements which are small but not small enough to allow for deflation. The idea at this point is to begin the QR algorithm deflating at one of the two small elements. The reason is that when we apply a givens rotation in effect we are multiplying the two small elements together and therefore they end up becoming smaller than the threshold for deflation. Aggressive deflation is due to Francis 1961 [1, 2] but Watkins in 1995 [21] proved the stability of this method.

5.6 Balancing

Balancing is a technique when the elements in a matrix vary significantly in size. The elements in the matrix A is multiplied by a matrix $D = diag(\rho, 1, 1, ..., 1)$ and by D^{-1}. Therefore DAD^{-1} divides by ρ the first column and multiplies by ρ the first row. We thus choose ρ such that after the multiplication all elements are balanced

$$\rho \sum_{j \neq 1} |a_{1j}| = \frac{1}{\rho} \sum_{i \neq 1} |a_{i1}|. \tag{13}$$

The balancing algorithm is due to Osborne 1960 [10].

5.7 Graded matrices

A matrix is graded when it shows a systematic decrease or increase in the values of the elements. It is observed that in general the QR algorithm succeeds when the matrix is graded downward but it fails when the matrix is graded up-wards. Therefore balancing can help and solve the problem.

6 The Implicitly shifted QR algorithm

The explicit QR algorithm computes the general Schur decomposition for a complex matrix. However, complex arithmetic is expensive and for this reason it is preferable to determine the "real" Schur decomposition. With this technique we move all the complex arithmetics to the eigenvectors and we leave the QR algorithm in real form. The real Schur form replaces the triangular matrix T with a quasi-triangular matrix T with 2×2 blocks along the diagonal. The blocks contain pairs of complex conjugate eigenvalues. This is achieved through a technique called "implicit shifting". The implicit shifted QR algorithm was first proposed by Francis [1, 2].

6.1 The double shift and the implicit Q theorem

Starting from a real matrix A we can reduce it to upper Hessenberg form via Householder reflections. H will still be real. We then apply the shifted QR algorithm with Wilkinson shift. When the Wilkinson shift finds a pair of complex eigenvalues we apply the following double shift

$$QR = (H - \kappa I)(H - \bar{\kappa} I). \tag{14}$$

Notice that $(H - \kappa I)(H - \bar{\kappa} I) = H^2 - 2Re(\kappa)H + |\kappa|^2 I$ is a real matrix. Therefore Q and R are also real matrices. Notice that this method is called the Francis double shift strategy.

Notice that it requires $O(n^3)$ operation to conduct H^2 and therefore the method appears to be unsatisfactory. However the implicit Q theorem can be used to avoid performing these computations.

The implicit Q theorem states that for a matrix A of order n, let $H = Q^* A Q$ be a reduction of A to Hessenberg form. If the elements in the lower diagonal of H are non-zero (that is the H is un-reduced) then Q and H are uniquely determined by the first or last column of Q.

Intuitively we can reason why this theorem is true. In order to reduce a matrix into upper Hessenberg form we need to make $(n-1)(n-2)/2$ elements equal to zero. A unitary matrix Q has $n(n-1)/2$ degrees of freedom. This leaves $(n-1)$ free degrees which can be used to determine the first column of Q.

Note that the proof of the theorem is an orthogonalization algorithm for computing Q and H and goes under the name of the Arnoldi method.

6.2 The implicit double shift QR algorithm

We can use the result of the implicit Q theorem to design the implicit double shift QR algorithm.

1. We determine the first column c of $C = H^2 - 2Re(\kappa)H + |\kappa|^2 I$.
2. Next let Q_0 be a Householder transformation such that $Q_0^* c = \sigma e_1$.
3. Next let $H_1 = Q_0^* A Q_0$.
4. Use Householder transformations to reduce H_1 into upper Hessenberg form and call it $\hat{H}$. Call Q_1 the accumulated transformations.
5. Set $\hat{Q} = Q_0 Q_1$.

To show that this algorithm works we need to show that $\hat{Q}$ and $\hat{H}$ are the matrices that one would have obtained using the explicit QR algorithm.

First perform the QR decomposition of the matrix C as follows

$$(cC_p) = (qQ_p) \begin{pmatrix} \rho & r \\ 0 & R_p \end{pmatrix}. \tag{15}$$

Notice that the Q matrix in the QR decomposition of C is the matrix that we want to show to be equal to $\hat{Q}$. We notice that $c = q\rho$. We also have the $c = \sigma q^{(0)}$ where $q^{(0)}$

is the first column of Q_0. This shows that the first column of Q_0 and the first column of $\hat{Q}$ are up to scaling the same. Therefore by the implicit Q theorem which states the uniqueness of the reduction to upper Hessenberg once the first column of Q is chosen we conclude that $\hat{H}$ is the matrix that would have resulted from the QR algorithm and $\hat{Q}$ is the Q that would have resulted from using the QR algorithm.

In the above formulation the only expensive steps are the computation of the first column of C and the reduction to upper Hessenberg form of H_1 to obtain $\hat{H}$.

The computation of c takes $O(1)$ operations because only the first three components of c are non-zero and we can find the formula for these components quite rapidly.

The reduction of H_1 to upper Hessenberg can be done in $O(n^2)$ steps since the Householder reflection Q_0 multiplying A which is in upper Hessenberg form itself leaves a matrix that with only two elements below the sub-diagonal different from zero.

Notice that the shifts need not be a conjugate pair but we could have any number of shifts

7 The Generalized eigenvalue problem

The generalized eigenvalue problem determines non-trivial solutions to the system

$$Ax = \lambda Bx. \tag{16}$$

Notice that if $B = I$ the problem reduces to the regular eigenvalue problem. The generalized eigenvalue problem can have an infinite number of eigenvalues. We can find an equivalent Hessenberg and Schur form and the adaptation of the QR algorithm goes under the name of the QZ algorithm.

7.1 Theoretical background

We call the pair (x, λ) a right eigenpair of the pencil (A, B) if $Ax = \lambda Bx$. λ is called the eigenvalue of the pencil and x is called the eigenvector of the pencil. From a geometric point of view this means that the direction of Ax and Bx is the same unlike regular eigenvectors.

The pencil (A, B) is regular if $\det(A - \lambda B)$ is not identically zero. A regular pencil can only have a finite number of eigenvalues.

Just as we use similarity transformations to preserve the eigenstructure we can use the pair of non-singular matrices (U, V) to preserve the structure of the pencil. We call the pencil (U^*AV, U^*BV) equivalent to (A, B).

There exist a pair of U and V such that $S = U^*AV$ and $T = U^*BV$ are triangular for the regular pencil (A, B). This is called the generalized Schur form.

The eigenvalues of the pencil (A, B) are denoted by the pair $(\alpha_{ii}, \beta_{ii})$. If $\beta_{ii} \neq 0$ then $\lambda = \alpha_{ii}/\beta_{ii}$ is an eigenvalue of A. Otherwise the eigenvalue is infinite.

If B is non-singular then the generalized eigenvalue problem becomes a regular eigenvalue problem since $B^{-1}Ax = \lambda x$. The reduction is not possible when B is singular. If B is also ill conditioned we can use shifts in the pencil of the form $Ax = \mu(B + wA)x$ where the eigenpair is now $(\lambda/(1 + w\lambda), x)$. Perhaps now $B + wA$ is well conditioned.

7.2 The QZ algorithm

Before we describe the QZ algorithm we reduce the matrix to a more convenient form which for the generalized eigenvalue problem is the Hessenber-Triangular form.

Just as we did with the implicit QR algorithm we can find U and V such that the regular pencil (A, B) is reduced to real Schur form. This means that $S = U^*AV$ and $T = U^*BV$ are quasi-triangular matrices.

We now reduce the matrix to Hessenberg-triangular form. The procedure applied to a regular pencil (A, B) reduces A to upper Hessenberg form and B to triangular form.

Notice that if B is triangular then B^{-1} is lower triangular and $B^{-1}A$ is upper Hessenberg. Therefore the Hessenberg-Triangular form corresponds to reducing $B^{-1}A$ to upper-Hessenberg form.

First we form the matrix of Householder transformations Q such that Q^TB is upper triangular. We also perform the operation Q^TA. Next we use plane rotations to reduce A into upper-Hessenberg form while preserving the triangularity of B. This step is done very carefully. For example let $n = 5$ and suppose we want to eliminate the a_{51} element of A. We use rotations that involve rows 4 and 5. This causes the element $b_{5,4}$ to become non-zero. We then rotate again to return the element $b_{5,4} = 0$ and this maintains $a_{5.1} = 0$.

We are now ready to discuss the QZ algorithm. This is also an iterative reduction of a real Hessenberg-triangular pencil to a real generalized Schur form.

Suppose (A, B) is a pencil in Hessenberg-Tridiagonal form. The main idea is to use the doubly shifted implicit QR on the matrix $C = AB^{-1}$ which is an upper Hessenberg matrix. This is the procedure for the algorithm

- Find the first column of C.
- Let H be the Householder transformation such that $Q_0^*c = \sigma e_1$.
- Find Q and Z such that $Q^*(HA, HB)Z$ is again in Hessenberg-Triangular form.

This algorithm works because $\hat{C} = Q^*H^*CHQ$, therefore we are applying a double step of the implicit QR algorithm on C. For this reason elements in the sub-diagonal of C will converge to zero. This means that either A is becoming triangular or if the elements or b become zero then we have found an infinite eigenvalue and we can deflate the problem.

The QZ algorithm was introduced by Moler and Stewart in 1973 [8].

References

[1] J. G. F. Francis, *The qr transformation part i*, Computer Journal **4** (1961), 135–148.

[2] ———, *The qr transformation part ii*, Computer Journal **4** (1962), 332–345.

[3] H. Hotelling, *Analysis of a complex statistical data into principal components part i*, Journal of Educational Psychology **24** (1933), 417–441.

[4] ———, *Analysis of a complex statistical data into principal components part ii*, Journal of Educational Psychology **24** (1933), 498–520.

[5] A. S. Householder and F. L. Bauer, *On certain methods for expanding the characteristic polynomial*, Numerische Mathematik **1** (1958), 29–37.

[6] V. N. Kublanovskaya, *On some algorithms for the solution of the complete eigenvalue problem*, USSR Computational Mathematics and Mathematical Physics **1** (1961), 637–657.

[7] C. B. Moler and G. W. Stewart, *Studies in parical mathematics, ii: The evaluation of the latent roots and latent vectors of a matrix*, Proceedings of the Royal Society of Endiburgh **57** (1937), 269–304.

[8] ———, *An algorithm for generalized matrix eigenvalue problems*, SIAM Journal on Numerical Aanlysis **10** (1973), 241–256.

[9] C. L. Muntz, *Solution direct de l'equation seculaire et de quelques problems analogues transcendents*, Comptus Rendus de l'Academie des Sciences **156** (1913), 43–46.

[10] E. E. Osborne, *On pre-conditioning of matrices.*

[11] A. M. Ostrowski, *On the convergence of the rayleigh quotient iteration for the computation of the characteristic roots and vectors. part i*, Archives for Rational Mechanics and Analysis **1** (1958), 233–241.

[12] ———, *On the convergence of the rayleigh quotient iteration for the computation of the characteristic roots and vectors. part ii*, Archives for Rational Mechanics and Analysis **2** (1958), 423–428.

[13] ———, *On the convergence of the rayleigh quotient iteration for the computation of the characteristic roots and vectors. part iii*, Archives for Rational Mechanics and Analysis **3** (1959), 325–340.

[14] ———, *On the convergence of the rayleigh quotient iteration for the computation of the characteristic roots and vectors. part iv*, Archives for Rational Mechanics and Analysis **3** (1959), 341–347.

[15] ———, *On the convergence of the rayleigh quotient iteration for the computation of the characteristic roots and vectors. part v*, Archives for Rational Mechanics and Analysis **3** (1959), 472–481.

[16] ———, *On the convergence of the rayleigh quotient iteration for the computation of the characteristic roots and vectors. part vi*, Archives for Rational Mechanics and Analysis **4** (1959), 153–165.

[17] H. Rutishauser, *Une methode pour la determination des valeurs propres d'une matrice*, Coomptes Rendus de l'Academie des Sciences **240** (1955), 34–36.

[18] ———, *Solution of eigenvalue problems with the lr-transformation*, National Bureau of Standards Applied Mathematics Series **49** (1958), 47–81.

[19] G. W. Stewart, *Error and perturbation bounds for subspaces associated with certain eigenvalue problems*, SIAM Review **15** (1973), 727–764.

[20] ———, *Matrix algorithms, volume ii: Eigensystems*, SIAM, Philadelphia, 2001.

[21] D. S. Watkins, *Forward stability and transmission of shifts in teh qr algorithm*, SIAM Journal on Matrix Analysis and Applications **16** (1995), 469–487.

[22] H Wielandt, *Beitrage zur mathematischen behandlung koomplexer eigenwertprobleme v: Bestimmung hoherer eigenwerte durch gebrochene iteration*, Gottinger, Germany, 1944.

[23] J. H. Wilkinson, *Householder's method for the solution of the algebraic eigenvalue problem*, Computer Journal **3** (1960), 23–27.

[24] ———, *The algebraic eigenvalue problem*, Clarendon Press, Oxford, 1965.

[25] ———, *Convergence of the lr, qr, and related algorithms*, Computer Journal **8** (1965), 77–84.

[26] ———, *Global convergence of tridiagonal qr algorithm with origin shifts*, Linear Algebra and its Applications **1** (1968), 409–420.