§9.4 The QR Algorithm

The QR factorization of a matrix: For a matrix (a_{ij}), let $x_1 = (a_{11}, \ldots, a_{n1})^t$ and $y_1 = (\|x_1\|_2, 0, \ldots, 0)^t$, then there is a Householder matrix P_1 such that $P_1 x_1 = y_1$ and

$$P_1 A = \begin{bmatrix} \|x_1\|_2 & a_{12}^{(1)} & \ldots & a_{1n}^{(1)} \\ 0 & a_{22}^{(1)} & \ldots & a_{2n}^{(1)} \\ \vdots & \vdots & \ldots & \vdots \\ 0 & a_{n2}^{(1)} & \ldots & a_{nn}^{(1)} \end{bmatrix}$$

Let $x_2 = (0, a_{22}^{(1)} \ldots, a_{n1}^{(1)})^t$ and $y_2 = (0, \|x_2\|_2, 0, \ldots, 0)^t$, then there is a Householder matrix P_2 such that $P_2 x_2 = y_2$ and

$$P_2 P_1 A = \begin{bmatrix} \|x_1\|_2 & a_{12}^{(1)} & a_{13}^{(1)} & \ldots & a_{1n}^{(1)} \\ 0 & \|x_2\|_2 & a_{23}^{(2)} & \ldots & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(2)} & \ldots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & a_{n3}^{(2)} & \ldots & a_{nn}^{(2)} \end{bmatrix}.$$

If we continue this procedure, we will have Householder matrices $P_3, \ldots, P_{n-1}$ such that

$$P_{n-1} \cdots P_2 P_1 A = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \ldots & r_{1n} \\ 0 & r_{22} & r_{23} & \ldots & r_{2n} \\ 0 & 0 & r_{33} & \ldots & r_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & r_{nn} \end{bmatrix} = R,$$

an upper triangular matrix. Since all P_i's are symmetric and orthogonal, $P_i^{-1} = P_i^t = P_i$. Let $Q = P_1 P_2 \cdots P_{n-1}$. Then,

$$A = QR.$$

Note that Q is still orthogonal (but not symmetric).

The operation count for the QR factorization is $O(n^3)$.

Remark: For a system $Ax = b$, if we apply the QR factorization to the matrix $[A \,|\, b]$, say $Q[A \,|\, b] = [R \,|\, b']$, i.e.,

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & \ldots & r_{1n} & b_1' \\ 0 & r_{22} & r_{23} & \ldots & r_{2n} & b_2' \\ 0 & 0 & r_{33} & \ldots & r_{3n} & b_3' \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & r_{nn} & b_n' \end{bmatrix}$$

Then solving $Ax = b$ is equivalent to solving $Rx = b'$. You can use backward-substitution method to solve $Rx = b'$ as we did in the last step of Gaussian elimination method. This is the QR method for solving linear systems. This method is much more stable than Gaussian elimination.

The QR method for eigenvalues of tridiagonal matrices:

The basics:

(1) If A is upper Hessenberg and symmetric, then A is tridiagonal.

(2) The product of two upper-Hessenberg matrices is still upper-Hessenberg. (in your homework)

(3) If $w = (0, \ldots, 0, w_k, w_{k+1}, 0, \ldots, 0)^t$, then $P = (p_{ij}) = I - 2ww^t$ is tridiagonal (only off-diagonal elements $p_{k,k+1}$ and $p_{k+1,k}$ may not be zero), and thus upper-Hessenberg.

(4) The eigenvalues of a diagonal matrix are the diagonal elements of the matrix.

(5) If A is a tridiagonal matrix with one of the off-diagonal element, say $a_{k+1,k}$, is zero or smaller than the tolerance given, then we can split A into two smaller matrices (one from the first k rows and first k columns of A, the another from the last $n-k$ rows and last $n-k$ columns of A), and the eigenvalues of these two smaller matrices are the eigenvalues of A.

Let A be a symmetric tridiagonal matrix with no off-diagonal elements zero or small. We want to find its eigenvalues.

If we apply the method of QR factorization above, there are Householder matrices $P_1, P_2, \ldots, P_{n-1}$ and an upper-triangular matrix R such that

$$P_{n-1} \cdots P_2 P_1 A = R^{(1)}.$$

Note that P_i's are all tridiagonal from (3) in the basics. P_i^t's are still tridiagonal. So,

$$A = (P_1^t P_2^t \cdots P_{n-1}^t) R = Q^{(1)} R^{(1)}$$

since P_i's are all orthogonal ($P_i^t = P_i^{-1}$), where

$$Q^{(1)} = P_1^t P_2^t \cdots P_{n-1}^t.$$

Let $A^{(1)} = R^{(1)} Q^{(1)}$. Then

(a) $A^{(1)} = R^{(1)} Q^{(1)} = (Q^{(1)})^t A Q^{(1)}$.

So, $A^{(1)}$ is symmetric and similar to A (this requires that $Q^{(1)}$ is orthogonal!).

(b) $A^{(1)}$ is still tridiagonal since $R^{(1)}$ and $Q^{(1)}$ are upper-Hessenberg and $A^{(1)}$ is symmetric.

A result: The absolute values of the off-diagonal elements of $A^{(1)}$ are usually smaller than the ones of A. (We can not prove it in this class)

If we continue this procedure, that is, perform the following iterations:

For $k = 1, 2, 3, \ldots$, do

$$A^{(k)} = Q^{(k+1)} R^{(k+1)}, \qquad A^{(k+1)} = R^{(k+1)} Q^{(k+1)}.$$

Then $A^{(k)}$ will converges to a diagonal matrix D when k increases.

The iterations can be stopped when the off-diagonal elements are smaller than the tolerance given.

The diagonal elements of D are the eigenvalues of A since $A^{(k)}$'s are similar to A.

This is the basic idea of QR method for finding eigenvalues of matrices.

Two simple ways to improve the method:

(a) Replace the Householder matrices by rotation matrices to reduce operation counts.

(b) Use shifting technique to speed up the convergence of the off-diagonal elements to 0.

For (a): the basic operation in the Householder method for the tridiagonal matrices is to change

$$x = (x_1, \ldots, x_{k-1}, x_k, x_{k+1}, 0 \ldots, 0)^t$$

to

$$y = (x_1, \ldots, x_{k-1}, *, 0, 0 \ldots, 0)^t.$$

It can be done if we know how to simply find an orthogonal matrix to change $a = (a_1, a_2)^t$ to $(\|a\|_2, 0)^t$. Let θ be the angle between these two vectors in the plane. Then

$$\cos\theta = \frac{a_1}{\|a\|_2}, \qquad \sin\theta = \frac{a_2}{\|a\|_2}$$

and

$$\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \|a\|_2 \\ 0 \end{bmatrix}$$

To change

$$x = (x_1, \ldots, x_{k-1}, x_k, x_{k+1}, 0 \ldots, 0)^t$$

to

$$y = (x_1, \ldots, x_{k-1}, *, 0, 0 \ldots, 0)^t,$$

we simply define

$$\cos\theta = \frac{x_k}{\|(x_k, x_{k+1})\|_2}, \qquad \sin\theta = \frac{x_{k+1}}{\|(x_k, x_{k+1})\|_2}$$

and

$$P = \begin{bmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ & & & \cos\theta & \sin\theta & & & \\ & & & -\sin\theta & \cos\theta & & & \\ & & & & & 1 & & \\ & & & & & & \cdots & \\ & & & & & & & 1 \end{bmatrix}$$

with the only nonzero off-diagonal elements $p_{k+1,k}$ and $p_{k,k+1}$. Then $Px = y$. This kind of matrices are called **rotation matrices**. They are orthogonal (we need this since we only can perform "similar" transformations on A!), but not symmetric.

Remark: These matrices are a little easier to construct than the Householder's. The saving is small (save 4 $\times/\div$ operations). But the saving will be big if you need to construct this kind of matrices millions times. This is the case for the QR method for eigenvalues.

For (b): If A has distinct eigenvalues λ_i's with $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|$, the rate of convergence of the off-diagonal element $a^{(k)}_{i+1,i}$ depends on the ratio λ_{i+1}/λ_i.

The idea is to introduce a shifting to make one of the ratio very small, usually the last ratio λ_n/λ_{n-1}. In this way, $a^{(k)}_{n,n-1}$ goes to zero much faster than other off-diagonal elements. So, "choose"

a number s_k which is close to the eigenvalue to which $a_{nn}^{(k)}$ will converge, say λ_n, then the matrix $A^{(k)} - s_k I$ has eigenvalues $\lambda_1 - s_k, \ldots, \lambda_n - s_k$, and the ratio

$$\frac{\lambda_n - s_k}{\lambda_{n-1} - s_k}$$

will be very small. Then continue the iterations on the shifted matrix $A^{(k)} - s_k I$. Let

$$A^{(k)} - s_k I = Q^{(k)} R^{(k)}.$$

Define

$$A^{(k+1)} = R^{(k)} Q^{(k)} + s_k I.$$

The same shifting technique can be used on $A^{(k+1)}$. Note that $A^{(k)}$ and $A^{(k+1)}$ have same eigenvalues.

At step k, s_k is chosen to be the eigenvalue of

$$\begin{bmatrix} a_{n-1,n-1}^{(k)} & a_{n,n-1}^{(k)} \\ a_{n,n-1}^{(k)} & a_{n,n}^{(k)} \end{bmatrix},$$

which is the closest to $a_{n,n}^{(k)}$.

When $|a_{n,n-1}^{(k)}|$ is less than the tolerance given, then $a_{n,n}^{(k)}$ is a numerical eigenvalue of the shifted matrix $A^{(k)}$. If we add the previous shifting value back to $a_{n,n}^{(k)}$, we get a numerical eigenvalue of A. At the same time, we can split the final shifted matrix $A^{(k)}$ into two smaller matrices (one is 1×1) as explained in (4) of the basics. Then continue the iterations on the smaller matrix.

The method will be stopped when it reaches a 2×2 matrix whose eigenvalues are easily obtained by solving its characteristic polynomial.

An example is given on page 581 (you need the definitions of the constants in Alg. 9.6 for the example).

There are much much more about QR method for calculating eigenvalues!