Chapter 5
QR Decomposition

One of the main themes of this book is decomposition of matrices to compact (e.g., triangular or diagonal) form by orthogonal transformations. We will now introduce the first such decomposition, the *QR decomposition*, which is a factorization of a matrix A in a product of an orthogonal matrix and a triangular matrix. This is more ambitious than computing the LU decomposition, where the two factors are both required only to be triangular.

5.1 Orthogonal Transformation to Triangular Form

By a sequence of Householder transformations[7] we can transform any matrix $A \in \mathbb{R}^{m\times n}$, $m \geq n$,

$$A \longrightarrow Q^T A = \begin{pmatrix} R \\ 0 \end{pmatrix}, \qquad R \in \mathbb{R}^{n\times n},$$

where R is upper triangular and $Q \in \mathbb{R}^{m\times m}$ is orthogonal. The procedure can be conveniently illustrated using a matrix of small dimension. Let $A \in \mathbb{R}^{5\times 4}$. In the first step we zero the elements below the main diagonal in the first column,

$$H_1 A = H_1 \begin{pmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{pmatrix} = \begin{pmatrix} + & + & + & + \\ 0 & + & + & + \\ 0 & + & + & + \\ 0 & + & + & + \\ 0 & + & + & + \end{pmatrix},$$

where +'s denote elements that have changed in the transformation. The orthogonal matrix H_1 can be taken equal to a Householder transformation. In the second step we use an embedded Householder transformation as in (4.3) to zero the elements

[7] In this chapter we use Householder transformations, but analogous algorithms can be formulated in terms of plane rotations.

below the main diagonal in the second column:

$$H_2 \begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{pmatrix} = \begin{pmatrix} \times & \times & \times & \times \\ 0 & + & + & + \\ 0 & 0 & + & + \\ 0 & 0 & + & + \\ 0 & 0 & + & + \end{pmatrix}.$$

Again, on the right-hand side, +'s denote elements that have been changed in the transformation, and ×'s denote elements that are unchanged in the present transformation.

In the third step we annihilate elements below the diagonal in the third column:

$$H_3 \begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \end{pmatrix} = \begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & + & + \\ 0 & 0 & 0 & + \\ 0 & 0 & 0 & + \end{pmatrix}.$$

After the fourth step we have computed the upper triangular matrix R. The sequence of transformations is summarized

$$Q^T A = \begin{pmatrix} R \\ 0 \end{pmatrix}, \qquad Q^T = H_4 H_3 H_2 H_1.$$

Note that the matrices H_i have the following structure (here we assume that $A \in \mathbb{R}^{m \times n}$):

$$\begin{aligned} H_1 &= I - 2u_1 u_1^T, \quad u_1 \in \mathbb{R}^m, \\ H_2 &= \begin{pmatrix} 1 & 0 \\ 0 & P_2 \end{pmatrix}, \quad P_2 = I - 2u_2 u_2^T, \quad u_2 \in \mathbb{R}^{m-1}, \\ H_3 &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & P_3 \end{pmatrix}, \quad P_3 = I - 2u_3 u_3^T, \quad u_3 \in \mathbb{R}^{m-2}, \end{aligned} \tag{5.1}$$

etc. Thus we embed the Householder transformations of successively smaller dimension in identity matrices, and the vectors u_i become shorter in each step. It is easy to see that the matrices H_i are also Householder transformations. For instance,

$$H_3 = I - 2u^{(3)} {u^{(3)}}^T, \quad u^{(3)} = \begin{pmatrix} 0 \\ 0 \\ u_3 \end{pmatrix}.$$

The transformation to triangular form is equivalent to a decomposition of the matrix A.

Theorem 5.1 (QR decomposition). *Any matrix $A \in \mathbb{R}^{m\times n}$, $m \geq n$, can be transformed to upper triangular form by an orthogonal matrix. The transformation is equivalent to a decomposition*

$$A = Q \begin{pmatrix} R \\ 0 \end{pmatrix},$$

where $Q \in \mathbb{R}^{m\times m}$ is orthogonal and $R \in \mathbb{R}^{n\times n}$ is upper triangular. If the columns of A are linearly independent, then R is nonsingular.

Proof. The constructive procedure outlined in the preceding example can easily be adapted to the general case, under the provision that if the vector to be transformed to a unit vector is a zero vector, then the orthogonal transformation is chosen equal to the identity.

The linear independence of the columns of

$$\begin{pmatrix} R \\ 0 \end{pmatrix}$$

follows from Proposition 2.4. Since R is upper triangular, the linear independence implies that its diagonal elements are nonzero. (If one column had a zero on the diagonal, then it would be a linear combination of those to the left of it.) Thus, the determinant of R is nonzero, which means that R is nonsingular. ∎

We illustrate the QR decomposition symbolically in Figure 5.1.

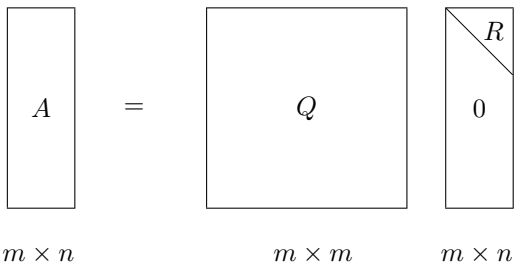

Figure 5.1. *Symbolic illustration of the QR decomposition.*

Quite often it is convenient to write the decomposition in an alternative way, where the only part of Q that is kept corresponds to an orthogonalization of the columns of A; see Figure 5.2.

This *thin QR decomposition* can be derived by partitioning $Q = (Q_1\ Q_2)$, where $Q_1 \in \mathbb{R}^{m\times n}$, and noting that in the multiplication the block Q_2 is multiplied by zero:

$$A = (Q_1\ Q_2) \begin{pmatrix} R \\ 0 \end{pmatrix} = Q_1 R. \tag{5.2}$$

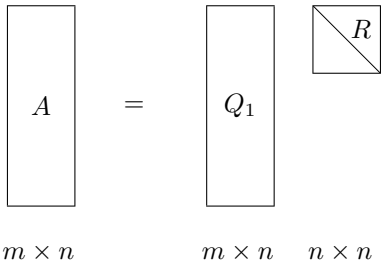

Figure 5.2. *Thin QR decomposition $A = Q_1R$.*

It is seen from this equation that $\mathcal{R}(A) = \mathcal{R}(Q_1)$; thus we have now computed an *orthogonal basis* of the range space $\mathcal{R}(A)$. Furthermore, if we write out column j in (5.2),

$$a_j = Q_1 r_j = \sum_{i=1}^{j} r_{ij} q_i,$$

we see that *column j in R holds the coordinates of a_j in the orthogonal basis.*

Example 5.2. We give a simple numerical illustration of the computation of a QR decomposition in MATLAB:

```
A =  1     1     1
     1     2     4
     1     3     9
     1     4    16
 >> [Q,R]=qr(A)

Q =-0.5000    0.6708    0.5000    0.2236
   -0.5000    0.2236   -0.5000   -0.6708
   -0.5000   -0.2236   -0.5000    0.6708
   -0.5000   -0.6708    0.5000   -0.2236

R =-2.0000   -5.0000  -15.0000
         0   -2.2361  -11.1803
         0         0    2.0000
         0         0         0
```

The thin QR decomposition is obtained by the command `qr(A,0)`:

```
>> [Q,R]=qr(A,0)

Q =-0.5000    0.6708    0.5000
   -0.5000    0.2236   -0.5000
   -0.5000   -0.2236   -0.5000
   -0.5000   -0.6708    0.5000
```

```
R =-2.0000   -5.0000  -15.0000
         0   -2.2361  -11.1803
         0         0    2.0000   ∎
```

5.2 Solving the Least Squares Problem

Using the QR decomposition, we can solve the least squares problem

$$\min_x \| b - Ax \|_2, \tag{5.3}$$

where $A \in \mathbb{R}^{m\times n}$, $m \geq n$, without forming the normal equations. To do this we use the fact that the Euclidean vector norm is invariant under orthogonal transformations (Proposition 4.8):

$$\|Qy\|_2 = \|y\|_2.$$

Introducing the QR decomposition of A in the residual vector, we get

$$\begin{aligned}\|r\|_2^2 = \|b - Ax\|_2^2 &= \left\| b - Q \begin{pmatrix} R \\ 0 \end{pmatrix} x \right\|_2^2 \\ &= \left\| Q(Q^T b - \begin{pmatrix} R \\ 0 \end{pmatrix} x) \right\|_2^2 = \left\| Q^T b - \begin{pmatrix} R \\ 0 \end{pmatrix} x \right\|_2^2.\end{aligned}$$

Then we partition $Q = (Q_1\ Q_2)$, where $Q_1 \in \mathbb{R}^{m\times n}$, and denote

$$Q^T b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} := \begin{pmatrix} Q_1^T b \\ Q_2^T b \end{pmatrix}.$$

Now we can write

$$\|r\|_2^2 = \left\| \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} - \begin{pmatrix} Rx \\ 0 \end{pmatrix} \right\|_2^2 = \|b_1 - Rx\|_2^2 + \|b_2\|_2^2. \tag{5.4}$$

Under the assumption that the columns of A are linearly independent, we can solve

$$Rx = b_1$$

and minimize $\|r\|_2$ by making the first term in (5.4) equal to zero. We now have proved the following theorem.

Theorem 5.3 (least squares solution by QR decomposition). *Let the matrix $A \in \mathbb{R}^{m\times n}$ have full column rank and thin QR decomposition $A = Q_1 R$. Then the least squares problem $\min_x \| Ax - b \|_2$ has the unique solution*

$$x = R^{-1} Q_1^T b.$$

Example 5.4. As an example we solve the least squares problem from the beginning of Section 3.6. The matrix and right-hand side are

```
A =  1     1              b = 7.9700
     1     2                 10.2000
     1     3                 14.2000
     1     4                 16.0000
     1     5                 21.2000
```

with thin QR decomposition and least squares solution

```
>> [Q1,R]=qr(A,0)      % thin QR

Q1 = -0.4472   -0.6325
     -0.4472   -0.3162
     -0.4472    0.0000
     -0.4472    0.3162
     -0.4472    0.6325

R = -2.2361   -6.7082
          0    3.1623

>> x=R\(Q1'*b)

x = 4.2360
    3.2260
```

Note that the MATLAB statement `x=A\b` gives the same result, using exactly the same algorithm. ∎

5.3 Computing or Not Computing Q

The orthogonal matrix Q can be computed at the same time as R by applying the transformations to the identity matrix. Similarly, Q_1 in the thin QR decomposition can be computed by applying the transformations to the partial identity matrix

$$\begin{pmatrix} I_n \\ 0 \end{pmatrix}.$$

However, in many situations we do not need Q explicitly. Instead, it may be sufficient to apply the same sequence of Householder transformations. Due to the structure of the embedded Householder matrices exhibited in (5.1), the vectors that are used to construct the Householder transformations for reducing the matrix A to upper triangular form can be stored below the main diagonal in A, in the positions that were made equal to zero. An extra vector is then needed to store the elements on the diagonal of R.

In the solution of the least squares problem (5.3) there is no need to compute Q at all. By adjoining the right-hand side to the matrix, we compute

$$\begin{pmatrix} A & b \end{pmatrix} \quad \rightarrow \quad Q^T \begin{pmatrix} A & b \end{pmatrix} = \begin{pmatrix} Q_1^T \\ Q_2^T \end{pmatrix} \begin{pmatrix} A & b \end{pmatrix} = \begin{pmatrix} R & Q_1^T b \\ 0 & Q_2^T b \end{pmatrix},$$

and the least squares solution is obtained by solving $Rx = Q_1^T b$. We also see that

$$\min_x \| Ax - b \|_2 = \min_x \left\| \begin{pmatrix} Rx - Q_1^T b \\ Q_2^T b \end{pmatrix} \right\|_2 = \| Q_2^T b \|_2.$$

Thus the norm of the optimal residual is obtained as a by-product of the triangularization procedure.

5.4 Flop Count for QR Factorization

As shown in Section 4.3, applying a Householder transformation to an $m \times n$ matrix to zero the elements of the first column below the diagonal requires approximately $4mn$ flops. In the following transformation, only rows 2 to m and columns 2 to n are changed (see (5.1)), and the dimension of the submatrix that is affected by the transformation is reduced by one in each step. Therefore the number of flops for computing R is approximately

$$4 \sum_{k=0}^{n-1} (m-k)(n-k) \approx 2mn^2 - \frac{2n^3}{3}.$$

Then the matrix Q is available in factored form, as a product of Householder transformations. If we compute explicitly the full matrix Q, then in step $k+1$ we need $4(m-k)m$ flops, which leads to a total of

$$4 \sum_{k=0}^{n-1} (m-k)m \approx 4mn \left(m - \frac{n}{2} \right).$$

It is possible to take advantage of structure in the accumulation of Q to reduce the flop count somewhat [42, Section 5.1.6].

5.5 Error in the Solution of the Least Squares Problem

As we stated in Section 4.4, Householder transformations and plane rotations have excellent properties with respect to floating point rounding errors. Here we give a theorem, the proof of which can be found in [50, Theorem 19.3].

Theorem 5.5. *Assume that $A \in \mathbb{R}^{m \times n}$, $m \geq n$, has full column rank and that the least squares problem $\min_x \| Ax - b \|_2$ is solved using QR factorization by Householder transformations. Then the computed solution $\hat{x}$ is the exact least squares solution of*

$$\min_x \| (A + \Delta A)\hat{x} - (b + \delta b) \|_2,$$

where

$$\| \Delta A \|_F \leq c_1 mn\mu \| A \|_F + O(\mu^2), \quad \| \delta b \|_2 \leq c_2 mn \| b \| + O(\mu^2),$$

and c_1 and c_2 are small constants.

It is seen that, in the sense of backward errors, the solution is as good as can be hoped for (i.e., the method is *backward stable*). Using the perturbation theory in Section 6.6 one can estimate the forward error in the computed solution. In Section 3.6 we suggested that the normal equations method for solving the least squares problem has less satisfactory properties in floating point arithmetic. It can be shown that that method is not backward stable, unless the matrix A is well-conditioned. The pros and cons of the two methods are nicely summarized in [50, p. 399]. Here we give an example that, although somewhat extreme, demonstrates that for certain least squares problems the solution given by the method of normal equations can be much less accurate than that produced using a QR decomposition; cf. Example 3.12.

Example 5.6. Let $\epsilon = 10^{-7}$, and consider the matrix

$$A = \begin{pmatrix} 1 & 1 \\ \epsilon & 0 \\ 0 & \epsilon \end{pmatrix}.$$

The condition number of A is of the order 10^7. The following MATLAB script

```
x=[1;1]; b=A*x;
xq=A\b;              % QR decomposition
xn=(A'*A)\(A'*b);    % Normal equations
[xq xn]
```

gave the result

```
1.00000000000000   1.01123595505618
1.00000000000000   0.98876404494382
```

which shows that the normal equations method is suffering from the fact that the condition number of the matrix A^TA is the square of that of A. ∎

5.6 Updating the Solution of a Least Squares Problem

In some applications the rows of A and the corresponding elements of b are measured in real time. Let us call one row and the element of b an *observation*. Every time an observation arrives, a new least squares solution is to be computed. If we were to recompute the solution from scratch, it would cost $O(mn^2)$ flops for each new observation. This is too costly in most situations and is an unnecessarily heavy computation, since the least squares solution can be computed by *updating* the QR decomposition in $O(n^2)$ flops every time a new observation is available. Furthermore, the updating algorithm does not require that we save the orthogonal matrix Q!

Assume that we have reduced the matrix and the right-hand side

$$\begin{pmatrix} A & b \end{pmatrix} \quad \rightarrow \quad Q^T \begin{pmatrix} A & b \end{pmatrix} = \begin{pmatrix} R & Q_1^T b \\ 0 & Q_2^T b \end{pmatrix}, \tag{5.5}$$

from which the least squares solution is readily available. Assume that we have not saved Q. Then let a new observation be denoted $(a^T\ \beta)$, where $a \in \mathbb{R}^n$ and β is a scalar. We then want to find the solution of the augmented least squares problem

$$\min_x \left\| \begin{pmatrix} A \\ a^T \end{pmatrix} x - \begin{pmatrix} b \\ \beta \end{pmatrix} \right\|. \tag{5.6}$$

In terms of the new matrix, we can write the reduction (5.5) in the form

$$\begin{pmatrix} A & b \\ a^T & \beta \end{pmatrix} \quad \rightarrow \quad \begin{pmatrix} Q^T & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} A & b \\ a^T & \beta \end{pmatrix} = \begin{pmatrix} R & Q_1^T b \\ 0 & Q_2^T b \\ a^T & \beta \end{pmatrix}.$$

Therefore, we can find the solution of the augmented least squares problem (5.6) if we reduce

$$\begin{pmatrix} R & Q_1^T b \\ 0 & Q_2^T b \\ a^T & \beta \end{pmatrix}$$

to triangular form by a sequence of orthogonal transformations. The vector $Q_2^T b$ will play no part in this reduction, and therefore we exclude it from the derivation.

We will now show how to perform the reduction to triangular form using a sequence of plane rotations. The ideas of the algorithm will be illustrated using a small example with $n = 4$. We start with

$$\begin{pmatrix} R & b_1 \\ a^T & \beta \end{pmatrix} = \begin{pmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \\ \times & \times & \times & \times & \times \end{pmatrix}.$$

By a rotation in the $(1, n+1)$ plane, we zero the first element of the bottom row vector. The result is

$$\begin{pmatrix} + & + & + & + & + \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \\ 0 & + & + & + & + \end{pmatrix},$$

where +'s denote elements that have been changed in the present transformation. Then the second element of the bottom vector is zeroed by a rotation in $(2, n+1)$:

$$\begin{pmatrix} \times & \times & \times & \times & \times \\ & + & + & + & + \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \\ & 0 & + & + & + \end{pmatrix}.$$

Note that the zero that was introduced in the previous step is not destroyed. After three more analogous steps, the final result is achieved:

$$\begin{pmatrix} \tilde{R} & \tilde{b}_1 \\ 0 & \tilde{\beta} \end{pmatrix} = \begin{pmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{pmatrix}.$$

A total of n rotations are needed to compute the reduction. The least squares solution of (5.6) is now obtained by solving $\tilde{R}x = \tilde{b}_1$.