Chapter 19
QR Factorization

Any orthogonal matrix can be written as the product of reflector matrices.
Thus the class of reflections is rich enough for all occasions
and yet each member is characterized by a single vector
which serves to describe its mirror.

— BERESFORD N. PARLETT, *The Symmetric Eigenvalue Problem* (1998)

A key observation for understanding the numerical properties of the
modified Gram–Schmidt algorithm is that it can be interpreted as
Householder QR factorization applied to the matrix A
augmented with a square matrix of zero elements on top.
These two algorithms are not only mathematically ...
but also numerically equivalent.
This key observation, apparently by Charles Sheffield,
was relayed to the author in 1968 by Gene Golub.

— ÅKE BJÖRCK, *Numerics of Gram-Schmidt Orthogonalization* (1994)

The great stability of unitary transformations in numerical analysis
springs from the fact that both the ℓ_2-norm
and the Frobenius norm are unitarily invariant.
This means in practice that even when rounding errors are made,
no substantial growth takes place in the
norms of the successive transformed matrices.

— J. H. WILKINSON,
Error Analysis of Transformations Based on the
Use of Matrices of the Form $I - 2ww^H$ (1965)

The QR factorization is a versatile computational tool that finds use in linear equation, least squares and eigenvalue problems. It can be computed in three main ways. The Gram–Schmidt process, which sequentially orthogonalizes the columns of A, is the oldest method and is described in most linear algebra textbooks. Givens transformations are preferred when A has a special sparsity structure, such as band or Hessenberg structure. Householder transformations provide the most generally useful way to compute the QR factorization. We explore the numerical properties of all three methods in this chapter. We also examine the use of iterative refinement on a linear system solved with a QR factorization and consider the inherent sensitivity of the QR factorization.

19.1. Householder Transformations

A Householder matrix (also known as a Householder transformation, or Householder reflector) is a matrix of the form

$$P = I - \frac{2}{v^T v} v v^T, \qquad 0 \neq v \in \mathbb{R}^n.$$

It enjoys the properties of symmetry and orthogonality, and, consequently, is involutary ($P^2 = I$). The application of P to a vector yields

$$Px = x - \left(\frac{2v^T x}{v^T v} \right) v.$$

Figure 19.1 illustrates this formula and makes it clear why P is sometimes called a Householder reflector: it reflects x about the hyperplane $\text{span}(v)^\perp$.

Householder matrices are powerful tools for introducing zeros into vectors. Consider the question, "Given x and y can we find a Householder matrix P such that $Px = y$?" Since P is orthogonal we clearly require that $\|x\|_2 = \|y\|_2$. Now

$$Px = y \quad \Longleftrightarrow \quad x - 2 \left(\frac{v^T x}{v^T v} \right) v = y,$$

and this last equation has the form $\alpha v = x - y$ for some α. But P is independent of the scaling of v, so we can set $\alpha = 1$.

With $v = x - y$ we have

$$v^T v = x^T x + y^T y - 2x^T y,$$

and, since $x^T x = y^T y$,

$$v^T x = x^T x - y^T x = \frac{1}{2} v^T v.$$

Therefore

$$Px = x - v = y,$$

as required. We conclude that, provided $\|x\|_2 = \|y\|_2$, we can find a Householder matrix P such that $Px = y$. (Strictly speaking, we have to exclude the case $x = y$, which would require $v = 0$, making P undefined.)

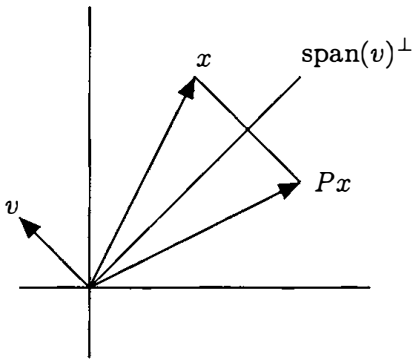

Figure 19.1. *Householder matrix P times vector x.*

Normally we choose y to have a special pattern of zeros. The usual choice is $y = \sigma e_1$ where $\sigma = \pm\|x\|_2$, which yields the maximum number of zeros in y. Then

$$v = x - y = x - \sigma e_1.$$

Most textbooks recommend using this formula with the sign chosen to avoid cancellation in the computation of $v_1 = x_1 - \sigma$:

$$\sigma = -\operatorname{sign}(x_1)\|x\|_2, \qquad v = x - \sigma e_1. \tag{19.1}$$

This is the approach used by the QR factorization routines in LINPACK [341, 1979] and LAPACK [20, 1999]. The prominence of the sign (19.1) has led to the myth that the other choice of sign is unsuitable. In fact, the other sign is perfectly satisfactory provided that the formula for v_1 is rearranged as follows [924, 1971], [291, 1976], [926, 1998, §6.3.1]:

$$\sigma = \operatorname{sign}(x_1)\|x\|_2, \tag{19.2a}$$

$$v_1 = x_1 - \sigma = \frac{x_1^2 - \|x\|_2^2}{x_1 + \sigma} = \frac{-(x_2^2 + \cdots + x_n^2)}{x_1 + \sigma}. \tag{19.2b}$$

For both choices of sign it is easy to show that $P = I - \beta vv^T$ with

$$\beta = \frac{2}{v^T v} = -\frac{1}{\sigma v_1}.$$

19.2. QR Factorization

A QR factorization of $A \in \mathbb{R}^{m\times n}$ with $m \geq n$ is a factorization

$$A = QR = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = Q_1 R_1,$$

where $Q \in \mathbb{R}^{m\times m}$ is orthogonal and $R_1 \in \mathbb{R}^{n\times n}$ is upper triangular. The matrix R is called upper trapezoidal, since the term triangular applies only to square

matrices. Depending on the context, either the full factorization $A = QR$ or the "economy size" version $A = Q_1 R_1$ can be called a QR factorization. A quick existence proof of the QR factorization is provided by the Cholesky factorization: if A has full rank and $A^T A = R^T R$ is a Cholesky factorization, then $A = AR^{-1} \cdot R$ is a QR factorization. The QR factorization is unique if A has full rank and we require R to have positive diagonal elements ($A = QD \cdot DR$ is a QR factorization for any $D = \mathrm{diag}(\pm 1)$).

The QR factorization can be computed by premultiplying the given matrix by a suitably chosen sequence of Householder matrices. The process is illustrated for a generic 4×3 matrix as follows:

$$A = \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} \xrightarrow{P_1} \left[\begin{array}{c|cc} \times & \times & \times \\ \hline 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{array}\right] \xrightarrow{P_2}$$

$$\left[\begin{array}{cc|c} \times & \times & \times \\ 0 & \times & \times \\ \hline 0 & 0 & \times \\ 0 & 0 & \times \end{array}\right] \xrightarrow{P_3} \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & 0 \end{bmatrix} = R.$$

The general process is adequately described by the kth stage of the reduction to triangular form. With $A_1 = A$ we have, at the start of the kth stage,

$$A_k = \begin{bmatrix} R_{k-1} & z_k & B_k \\ 0 & x_k & C_k \end{bmatrix}, \quad R_{k-1} \in \mathbb{R}^{(k-1)\times(k-1)}, \quad x_k \in \mathbb{R}^{m-k+1}, \tag{19.3}$$

where R_{k-1} is upper triangular. Choose a Householder matrix $\widetilde{P}_k$ such that $\widetilde{P}_k x_k = \sigma e_1$ and embed $\widetilde{P}_k$ into an $m \times m$ matrix

$$P_k = \begin{bmatrix} I_{k-1} & 0 \\ 0 & \widetilde{P}_k \end{bmatrix}. \tag{19.4}$$

Then let $A_{k+1} = P_k A_k$. Overall, we obtain $R = P_n P_{n-1} \dots P_1 A =: Q^T A$ ($P_n = I$ if $m = n$).

The Householder matrices P_i are never formed in practice; storage and computations use solely the Householder vector v. For example, to compute A_{k+1} we need to form $\widetilde{P} C_k$. We can write

$$\widetilde{P}_k C_k = (I - \beta v v^T) C_k = C_k - \beta v (v^T C_k), \qquad \beta = 2/(v^T v),$$

which shows that the matrix product can be formed as a matrix–vector product followed by an outer product. This approach is much more efficient than forming $\widetilde{P}_k$ explicitly and doing a matrix multiplication.

By taking σ nonnegative and switching between the formulae (19.1) and (19.2) according as x_1 is nonpositive and positive, respectively, we can obtain an R factor with nonnegative diagonal elements; this is done in [509, 1996, Algs. 5.1.1, 5.2.1], for example. However, this approach is not recommended for badly row scaled matrices, for reasons explained in §19.4.

The overall cost of the Householder reduction to triangular form is $2n^2(m-n/3)$ flops. Explicit formation of $Q = P_1P_2 \dots P_n$ can be done in two ways: from left to right or from right to left. The right to left evaluation is the more efficient, because the effective dimension of the intermediate products grows from $m-n$ to m, whereas with the left to right order it is m at each stage. The right to left evaluation requires $4(m^2n - mn^2 + n^3/3)$ flops, or $2n^2(m-n/3)$ flops (the same as for the reduction to triangular form) if only the first n columns of Q are formed. For most applications (such as solving the least squares problem) it suffices to leave Q in factored form.

19.3. Error Analysis of Householder Computations

It is well known that computations with Householder matrices are very stable. Wilkinson showed that the computation of a Householder vector, and the application of a Householder matrix to a given matrix, are both normwise stable, in the sense that the computed Householder vector is very close to the exact one and the computed update is the exact update of a tiny normwise perturbation of the original matrix [1233, 1965, pp. 153–162, 236], [1234, 1965]. Wilkinson also showed that the Householder QR factorization algorithm is normwise backward stable [1233, p. 236]. In this section we give a columnwise error analysis of Householder matrix computations. The columnwise bounds provide extra information over the normwise ones that is essential in certain applications (for example, the analysis of iterative refinement).

In the following analysis it is not worthwhile to evaluate the integer constants in the bounds explicitly, so we make frequent use of the notation

$$\tilde{\gamma}_k = \frac{cku}{1-cku}$$

introduced in (3.8), where c denotes a small integer constant.

Lemma 19.1. *Let $x \in \mathbb{R}^n$. Consider the following two constructions of $\beta \in \mathbb{R}$ and $v \in \mathbb{R}^n$ such that $Px = \sigma e_1$, where $P = I - \beta vv^T$ is a Householder matrix with $\beta = 2/(v^Tv)$:*

% "Usual" choice of sign, (19.1):	% Alternative choice of sign, (19.2):
% $\operatorname{sign}(\sigma) = -\operatorname{sign}(x_1)$.	% $\operatorname{sign}(\sigma) = \operatorname{sign}(x_1)$.
$v = x$	$v = x$
$s = \operatorname{sign}(x_1)\|x\|_2$ % $\sigma = -s$	$s = \operatorname{sign}(x_1)\|x\|_2$ % $\sigma = s$
$v_1 = v_1 + s$	Compute v_1 from (19.2)
$\beta = 1/(sv_1)$	$\beta = 1/(sv_1)$

In floating point arithmetic the computed $\widehat{\beta}$ and $\widehat{v}$ from both constructions satisfy $\widehat{v}(2\colon n) = v(2\colon n)$ and

$$\widehat{\beta} = \beta(1+\tilde{\theta}_n), \qquad \widehat{v}_1 = v_1(1+\tilde{\theta}_n),$$

where $|\tilde{\theta}_n| \le \tilde{\gamma}_n$.

Proof. We sketch the proof for the first construction. Each occurrence of δ denotes a different number bounded by $|\delta| \le u$. We compute $fl(x^Tx) = (1+\theta_n)x^Tx$, and then $fl(\|x\|_2) = (1+\delta)(1+\theta_n)^{1/2}(x^Tx)^{1/2} = (1+\theta_{n+1})\|x\|_2$ (the latter term $1+\theta_{n+1}$ is suboptimal, but our main aim is to keep the analysis simple). Hence $\widehat{s} = (1+\theta_{n+1})s$.

For notational convenience, define $w = v_1 + s$. We have $\widehat{w} = (v_1 + \widehat{s})(1+\delta) = w(1+\theta_{n+2})$ (essentially because there is no cancellation in the sum). Hence

$$\widehat{\beta} = fl(1/(\widehat{s}\widehat{w})) = \frac{(1+\delta)^2}{(1+\theta_{n+1})s(1+\theta_{n+2})w}$$
$$= \frac{(1+\delta)^2}{(1+\theta_{2n+3})sw} = (1+\theta_{4n+8})\beta.$$

The proof for the second construction is similar. □

For convenience we will henceforth write Householder matrices in the form $I - vv^T$, which requires $\|v\|_2 = \sqrt{2}$ and amounts to redefining $v := \sqrt{\beta}v$ and $\beta := 1$ in the representation of Lemma 19.1. We can then write, using Lemma 19.1,

$$\widehat{v} = v + \Delta v, \qquad |\Delta v| \le \tilde{\gamma}_m|v| \quad (v \in \mathbb{R}^m, \ \|v\|_2 = \sqrt{2}), \tag{19.5}$$

where, as required for the next two results, the dimension is now m.

The next result describes the application of a Householder matrix to a vector, and is the basis of all the subsequent analysis. In the applications of interest P is defined as in Lemma 19.1, but we will allow P to be an arbitrary Householder matrix. Thus v is an arbitrary, normalized vector, and the only assumption we make is that the computed $\widehat{v}$ satisfies (19.5).

Lemma 19.2. *Let $b \in \mathbb{R}^m$ and consider the computation of $y = \widehat{P}b = (I - \widehat{v}\widehat{v}^T)b = b - \widehat{v}(\widehat{v}^Tb)$, where $\widehat{v} \in \mathbb{R}^m$ satisfies* (19.5). *The computed $\widehat{y}$ satisfies*

$$\widehat{y} = (P + \Delta P)b, \qquad \|\Delta P\|_F \le \tilde{\gamma}_m,$$

where $P = I - vv^T$.

Proof. (Cf. the proof of Lemma 3.9.) We have

$$\widehat{w} := fl(\widehat{v}(\widehat{v}^Tb)) = (\widehat{v} + \Delta\widehat{v})(\widehat{v}^T(b + \Delta b)),$$

where $|\Delta\widehat{v}| \le u|\widehat{v}|$ and $|\Delta b| \le \gamma_m|b|$. Hence

$$\widehat{w} = (v + \Delta v + \Delta\widehat{v})(v + \Delta v)^T(b + \Delta b) =: v(v^Tb) + \Delta w,$$

where $|\Delta w| \le \tilde{\gamma}_m|v||v^T||b|$. Then

$$\widehat{y} = fl(b - \widehat{w}) = b - v(v^Tb) - \Delta w + \Delta y_1, \qquad |\Delta y_1| \le u|b - \widehat{w}|.$$

We have

$$|-\Delta w + \Delta y_1| \le u|b| + \tilde{\gamma}_m|v||v^T||b|.$$

Hence $\widehat{y} = Pb + \Delta y$, where $\|\Delta y\|_2 \le \tilde{\gamma}_m\|b\|_2$. But then $\widehat{y} = (P + \Delta P)b$, where $\Delta P = \Delta y b^T/b^Tb$ satisfies $\|\Delta P\|_F = \|\Delta y\|_2/\|b\|_2 \le \tilde{\gamma}_m$. □

Next, we consider a sequence of Householder transformations applied to a matrix. Again, each Householder matrix is arbitrary and need have no connection to the matrix to which it is being applied. In the cases of interest, the Householder matrices P_k have the form (19.4), and so are of ever-decreasing effective dimension, but to exploit this property would not lead to any significant improvement in the bounds. Since the P_j are applied to the columns of A, columnwise error bounds are to be expected, and these are provided by the next lemma.

We will assume that

$$r\tilde{\gamma}_m < \frac{1}{2}, \tag{19.6}$$

where r is the number of Householder transformations. We will write the jth columns of A and ΔA as a_j and Δa_j, respectively.

Lemma 19.3. *Consider the sequence of transformations*

$$A_{k+1} = P_k A_k, \qquad k = 1\colon r,$$

where $A_1 = A \in \mathbb{R}^{m\times n}$ and $P_k = I - v_k v_k^T \in \mathbb{R}^{m\times m}$ is a Householder matrix. Assume that the transformations are performed using computed Householder vectors $\widehat{v}_k \approx v_k$ that satisfy (19.5). *The computed matrix $\widehat{A}_{r+1}$ satisfies*

$$\widehat{A}_{r+1} = Q^T(A + \Delta A), \tag{19.7}$$

where $Q^T = P_r P_{r-1} \dots P_1$ and

$$\|\Delta a_j\|_2 \le r\tilde{\gamma}_m \|a_j\|_2, \qquad j = 1\colon n. \tag{19.8}$$

In the special case $n = 1$, so that $A \equiv a$, we have $\widehat{a}^{(r+1)} = (Q + \Delta Q)^T a$ with $\|\Delta Q\|_F \le r\tilde{\gamma}_m$.

Proof. The jth column of A undergoes the transformations $a_j^{(r+1)} = P_r \dots P_1 a_j$. By Lemma 19.2 we have

$$\widehat{a}_j^{(r+1)} = (P_r + \Delta P_r)\dots(P_1 + \Delta P_1)a_j, \tag{19.9}$$

where each ΔP_k depends on j and satisfies $\|\Delta P_k\|_F \le \tilde{\gamma}_m$. Using Lemma 3.7 we obtain

$$\begin{aligned} &\widehat{a}_j^{(r+1)} = Q^T(a_j + \Delta a_j), \\ &\|\Delta a_j\|_2 \le \left((1+\tilde{\gamma}_m)^r - 1\right)\|a_j\|_2 \le \frac{r\tilde{\gamma}_m}{1 - r\tilde{\gamma}_m}\|a_j\|_2 = r\tilde{\gamma}'_m\|a_j\|_2, \end{aligned} \tag{19.10}$$

using Lemma 3.1 and assumption (19.6). Finally, if $n = 1$, so that A is a column vector, then (as in the proof of Lemma 19.2) we can rewrite (19.7) as $\widehat{a}^{(r+1)} = (Q + \Delta Q)^T a$, where $\Delta Q^T = (Q^T \Delta a)a^T/a^T a$ and $\|\Delta Q\|_F = \|\Delta a\|_2/\|a\|_2 \le r\tilde{\gamma}_m$. □

Recall that columnwise error bounds are easily converted into normwise ones (Lemma 6.6). For example, (19.8) implies $\|\Delta A\|_F \le r\tilde{\gamma}_m \|A\|_F$.

Lemma 19.3 yields the standard backward error result for Householder QR factorization.

Theorem 19.4. *Let $\widehat{R} \in \mathbb{R}^{m\times n}$ be the computed upper trapezoidal QR factor of $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$ obtained via the Householder QR algorithm (with either choice of sign, (19.1) or (19.2)). Then there exists an orthogonal $Q \in \mathbb{R}^{m\times m}$ such that*

$$A + \Delta A = Q\widehat{R},$$

where

$$\|\Delta a_j\|_2 \leq \tilde{\gamma}_{mn}\|a_j\|_2, \qquad j = 1\colon n. \tag{19.11}$$

The matrix Q is given explicitly as $Q = (P_nP_{n-1}\dots P_1)^T$, where P_k is the Householder matrix that corresponds to the exact application of the kth step of the algorithm to $\widehat{A}_k$.

Proof. This is virtually a direct application of Lemma 19.3, with P_k defined as the Householder matrix that produces zeros below the diagonal in the kth column of the computed matrix $\widehat{A}_k$. One subtlety is that we do not explicitly compute the lower triangular elements of $\widehat{R}$, but rather set them to zero explicitly. However, it is easy to see that the conclusions of Lemmas 19.2 and 19.3 are still valid in these circumstances; the essential reason is that the elements of ΔPb in Lemma 19.2 that correspond to elements that are zeroed by the Householder matrix P are forced to be zero, and hence we can set the corresponding rows of ΔP to zero too, without compromising the bound on $\|\Delta P\|_F$. □

We note that for Householder QR factorization $\Delta P_k = 0$ for $k > j$ in (19.9), and consequently the factor $\tilde{\gamma}_{mn}$ in (19.11) can be reduced to $\tilde{\gamma}_{mj}$.

Theorem 19.4 is often stated in the weaker form $\|\Delta A\|_F \leq \tilde{\gamma}_{mn}\|A\|_F$ that is implied by (19.11) (see, e.g., [509, 1996, §5.2.1]). For a matrix whose columns vary widely in norm this normwise bound on ΔA is much weaker than (19.11). For an alternative way to express this backward error result define B by $A = BD_C$, where $D_C = \operatorname{diag}(\|A(:,j)\|_2)$; then the result states that there exists an orthogonal $Q \in \mathbb{R}^{m\times m}$ such that

$$(B + \Delta B)D_C = Q\widehat{R}, \qquad \|\Delta B(:,j)\|_2 \leq \tilde{\gamma}_{mn}, \tag{19.12}$$

so that $\|\Delta B\|_2/\|B\|_2 = O(u)$.

Note that the matrix Q in Theorem 19.4 is not computed by the QR factorization algorithm and is of purely theoretical interest. It is the fact that Q is exactly orthogonal that makes the result so useful. When Q is explicitly formed, two questions arise:

1. How close is the computed $\widehat{Q}$ to being orthonormal?
2. How large is $A - \widehat{Q}\widehat{R}$?

Both questions are easily answered using the analysis above.

We suppose that $Q = P_1P_2\dots P_n$ is evaluated in the more efficient right to left order. Lemma 19.3 gives (with $A_1 = I_m$)

$$\widehat{Q} = Q(I_m + \Delta I), \qquad \|\Delta I(:,j)\|_2 \leq \tilde{\gamma}_{mn}, \quad j = 1\colon n.$$

Hence

$$\|\widehat{Q} - Q\|_F \leq \sqrt{n}\,\tilde{\gamma}_{mn}, \tag{19.13}$$

showing that $\widehat{Q}$ is very close to an orthonormal matrix. Moreover, using Theorem 19.4,

$$\begin{aligned}\|(A-\widehat{Q}\widehat{R})(:,j)\|_2 &= \|(A-Q\widehat{R})(:,j)+((Q-\widehat{Q})\widehat{R})(:,j)\|_2\\ &\le \tilde{\gamma}_{mn}\|a_j\|_2+\|Q-\widehat{Q}\|_F\|\widehat{R}(:,j)\|_2\\ &\le \sqrt{n}\,\tilde{\gamma}_{mn}\|a_j\|_2.\end{aligned}$$

Thus if Q is replaced by $\widehat{Q}$ in Theorem 19.4, so that $A+\Delta A=\widehat{Q}\widehat{R}$, then the backward error bound remains true with an appropriate increase in the constant.

Finally, we consider use of the QR factorization to solve a linear system. Given a QR factorization of a nonsingular matrix $A\in\mathbb{R}^{n\times n}$, a linear system $Ax=b$ can be solved by forming Q^Tb and then solving $Rx=Q^Tb$. From Theorem 19.4, the computed $\widehat{R}$ is guaranteed to be nonsingular if $\kappa_2(A)n^{1/2}\tilde{\gamma}_{mn}<1$.

Theorem 19.5. *Let $A\in\mathbb{R}^{n\times n}$ be nonsingular. Suppose we solve the system $Ax=b$ with the aid of a QR factorization computed by the Householder algorithm. The computed $\widehat{x}$ satisfies*

$$(A+\Delta A)\widehat{x}=b+\Delta b,$$

where

$$\|\Delta a_j\|_2\le\tilde{\gamma}_{n^2}\|a_j\|_2,\quad j=1\colon n,\qquad \|\Delta b\|_2\le\tilde{\gamma}_{n^2}\|b\|_2.$$

Proof. By Theorem 19.4, the computed upper triangular factor $\widehat{R}$ satisfies $A+\Delta A=Q\widehat{R}$ with $\|\Delta a_j\|_2\le\tilde{\gamma}_{n^2}\|a_j\|_2$. By Lemma 19.3, the computed transformed right-hand side satisfies $\widehat{c}=Q^T(b+\Delta b)$, with $\|\Delta b\|_2\le\tilde{\gamma}_{n^2}\|b\|_2$. Importantly, the same orthogonal matrix Q appears in the equations involving $\widehat{R}$ and $\widehat{c}$.

By Theorem 8.5, the computed solution $\widehat{x}$ to the triangular system $\widehat{R}x=\widehat{c}$ satisfies

$$(\widehat{R}+\Delta R)\widehat{x}=\widehat{c},\qquad |\Delta R|\le\gamma_n|\widehat{R}|.$$

Premultiplying by Q yields

$$(A+\Delta A+Q\Delta R)\widehat{x}=b+\Delta b,$$

that is, $(A+\overline{\Delta A})\widehat{x}=b+\Delta b$, where $\overline{\Delta A}=\Delta A+Q\Delta R$. Using $\widehat{R}=Q^T(A+\Delta A)$ we have

$$\begin{aligned}\|\overline{\Delta a}_j\|_2 &\le \|\Delta a_j\|_2+\gamma_n\|\widehat{r}_j\|_2\\ &= \|\Delta a_j\|_2+\gamma_n\|a_j+\Delta a_j\|_2\\ &\le \tilde{\gamma}_{n^2}\|a_j\|_2.\qquad\square\end{aligned}$$

The proof of Theorem 19.5 naturally leads to a result in which b is perturbed. However, we can easily modify the proof so that only A is perturbed: the trick is to use the last part of Lemma 19.3 to write $\widehat{c}=(Q+\Delta Q)^Tb$, where $\|\Delta Q\|_F\le\tilde{\gamma}_{n^2}$, and to premultiply by $(Q+\Delta Q)^{-T}$ instead of Q in the middle of the proof. This leads to the result

$$(A+\Delta A)\widehat{x}=b,\qquad \|\Delta a_j\|_2\le\tilde{\gamma}_{n^2}\|a_j\|_2,\quad j=1\colon n. \tag{19.14}$$

An interesting application of Theorem 19.5 is to iterative refinement, as explained in §19.7.

19.4. Pivoting and Row-Wise Stability

It is natural to ask whether a small *row-wise* backward error bound holds for Householder QR factorization, since matrices A for which the rows vary widely in norm occur commonly in weighted least squares problems (see §20.8). In general, the answer is no. However, if column pivoting is used together with row pivoting or row sorting, and the choice of sign (19.1) is used in constructing the Householder vectors, then such a bound does hold. The column pivoting strategy exchanges columns at the start of the kth stage of the factorization to ensure that

$$\|a_k^{(k)}(k\colon m)\|_2 = \max_{j\geq k} \|a_j^{(k)}(k\colon m)\|_2. \tag{19.15}$$

In other words, it maximizes the norm of the active part of the pivot column.

Theorem 19.6 (Powell and Reid; Cox and Higham). *Let $\widehat{R} \in \mathbb{R}^{m\times n}$ be the computed upper trapezoidal QR factor of $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$ obtained via the Householder QR algorithm with column pivoting, with the choice of sign* (19.1). *Then there exists an orthogonal $Q \in \mathbb{R}^{m\times m}$ such that*

$$(A + \Delta A)\Pi = Q\widehat{R},$$

where Π is a permutation matrix that describes the overall effect of the column interchanges and

$$|\Delta a_{ij}| \leq j^2 \tilde{\gamma}_m \alpha_i \max_s |a_{is}|,$$

where

$$\alpha_i = \frac{\max_{j,k} |a_{ij}^{(k)}|}{\max_j |a_{ij}|}$$

and $A_k = (a_{ij}^{(k)})$. The matrix Q is defined as in Theorem 19.4. □

Problem 19.6 indicates the first part of a proof of Theorem 19.6 and gives insight into why column pivoting is necessary to obtain such a result.

Theorem 19.6 shows that the row-wise backward error is bounded by a multiple of $\max_i \alpha_i$. In general, the row-wise growth factors α_i can be arbitrarily large. The size of the α_i is limited by row pivoting and row sorting. With row pivoting, after the column interchange has taken place at the start of the kth stage we interchange rows to ensure that

$$|a_{kk}^{(k)}| = \max_{i\geq k} |a_{ik}^{(k)}|.$$

The alternative strategy of row sorting reorders the rows prior to carrying out the factorization so that

$$\|A(i,:)\|_\infty = \max_{j\geq i} \|A(j,:)\|_\infty, \qquad i = 1\colon m.$$

Note that row interchanges before or during Householder QR factorization have no mathematical effect on the result, because they can be absorbed into the Q factor and the QR factorization is essentially unique. The effect of row interchanges is to change the intermediate numbers that arise during the factorization, and hence to

Table 19.1. *Backward errors for QR factorization with no pivoting, row sorting, and column pivoting on matrix* (19.16).

Pivoting:	None	Row	Column	Row and column
Normwise (η)	2.9e-16	4.2e-16	3.2e-16	1.9e-16
Row-wise (η_R)	2.0e-4	2.7e-4	2.0e-4	4.0-16
$\max_i \alpha_i$	1.4e12	1.0e12	1.4e12	2.0e0

alter the effects of rounding errors. If row pivoting or row sorting is used it can be shown that $\alpha_i \le \sqrt{m}(1+\sqrt{2})^{n-1}$ for all i [275, 1998], [951, 1969], and experience shows that the α_i are usually small in practice. Therefore the α_i are somewhat analogous to the growth factor for GEPP. For the alternative choice of sign (19.2) in the Householder vectors, the α_i are unbounded even if row pivoting or row sorting is used and so row-wise stability is lost; see [275, 1998] for an illustrative example.

We give an example to illustrate the results. The matrix, from [1180, 1985], is

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & 1 \\ \mu & \mu & \mu \\ \mu & \mu & -\mu \end{bmatrix}. \tag{19.16}$$

We applied Householder QR factorization in MATLAB with $\mu = 10^{12}$, using both no pivoting and combinations of row sorting and column pivoting. The normwise and row-wise backward errors

$$\eta = \frac{\|A\Pi - \widehat{Q}\widehat{R}\|_2}{\|A\|_2}, \qquad \eta_R = \max_i \frac{\|(A\Pi - \widehat{Q}\widehat{R})(i,:)\|_2}{\|A(i,:)\|_2}$$

are shown in Table 19.1; here, $\widehat{Q}$ denotes the computed product of the Householder transformations. As expected, the computation is normwise backward stable in every case, but row-wise stability prevails only when both row sorting and column pivoting are used, with the size of the α_i being a good predictor of stability.

19.5. Aggregated Householder Transformations

In Chapter 13 we noted that LU factorization algorithms can be partitioned so as to express the bulk of the computation as matrix–matrix operations (level-3 BLAS). For computations with Householder transformations the same goal can be achieved by aggregating the transformations. This technique is widely used in LAPACK.

One form of aggregation is the "WY" representation of Bischof and Van Loan [117, 1987]. This involves representing the product $Q_r = P_r P_{r-1} \dots P_1$ of r Householder transformations $P_i = I - v_i v_i^T \in \mathbb{R}^{m\times m}$ (where $v_i^T v_i = 2$) in the form

$$Q_r = I + W_r Y_r^T, \qquad W_r, Y_r \in \mathbb{R}^{m\times r}.$$

This can be done using the recurrence

$$W_1 = -v_1, \quad Y_1 = v_1, \qquad W_i = [W_{i-1} \quad -v_i], \quad Y_i = [Y_{i-1} \quad Q_{i-1}^T v_i]. \tag{19.17}$$

Using the WY representation, a partitioned QR factorization can be developed as follows. Partition $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$ as

$$A = [A_1 \quad B], \qquad A_1 \in \mathbb{R}^{m\times r}, \tag{19.18}$$

and compute the Householder QR factorization of A_1,

$$P_r P_{r-1} \ldots P_1 A_1 = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}.$$

The product $P_r P_{r-1} \ldots P_1 = I + W_r Y_r^T$ is accumulated using (19.17), as the P_i are generated, and then B is updated according to

$$B \leftarrow (I + W_r Y_r^T)B = B + W_r(Y_r^T B),$$

which involves only level-3 BLAS operations. The process is now repeated on the last $m - r$ rows of B.

When considering numerical stability, two aspects of the WY representation need investigating: its construction and its application. For the construction, we need to show that $\widehat{Q}_r := I + \widehat{W}_r \widehat{Y}_r^T$ satisfies

$$\|\widehat{Q}_r \widehat{Q}_r^T - I\|_2 \leq d_1 u, \tag{19.19}$$

$$\|\widehat{W}_r\|_2 \leq d_2, \quad \|\widehat{Y}_r\|_2 \leq d_3, \tag{19.20}$$

for modest constants d_1, d_2, and d_3. Now

$$\widehat{Q}_i := I + \begin{bmatrix} \widehat{W}_{i-1} & -\widehat{v}_i \end{bmatrix} \begin{bmatrix} \widehat{Y}_{i-1}^T \\ fl(\widehat{v}_i^T \widehat{Q}_{i-1}) \end{bmatrix} = \widehat{Q}_{i-1} - \widehat{v}_i fl(\widehat{v}_i^T \widehat{Q}_{i-1}).$$

But this last equation is essentially a standard multiplication by a Householder matrix, $\widehat{Q}_i = (I - \widehat{v}_i \widehat{v}_i^T)\widehat{Q}_{i-1}$, albeit with less opportunity for rounding errors. It follows from Lemma 19.3 that the near orthogonality of $\widehat{Q}_{i-1}$ is inherited by $\widehat{Q}_i$; the condition on $\widehat{Y}_r$ in (19.20) follows similarly and that on $\widehat{W}_r$ is trivial. Note that the condition (19.19) implies that

$$\widehat{Q}_r = U_r + \Delta U_r, \quad U_r^T U_r = I, \quad \|\Delta U_r\|_2 \leq d_1 u, \tag{19.21}$$

that is, $\widehat{Q}$ is close to an exactly orthogonal matrix (see Problem 19.14).

Next we consider the application of $\widehat{Q}_r$. Suppose we form $C = \widehat{Q}_r B = (I + \widehat{W}_r \widehat{Y}_r^T)B$ for the B in (19.18), so that

$$\widehat{C} = fl\big(B + fl(\widehat{W}_r(\widehat{Y}_r^T B))\big).$$

Analysing this level-3 BLAS-based computation using (19.21) and the very general assumption (13.4) on matrix multiplication (for the 2-norm), it is straightforward to show that

$$\begin{aligned} \widehat{C} &= U_r B + \Delta C = U_r(B + U_r^T \Delta C), \\ \|\Delta C\|_2 &\leq \big[1 + d_1 + d_2 d_3(1 + c_1(r, m, n-r) \\ &\qquad + c_1(m, r, n-r))\big] u \|B\|_2 + O(u^2). \end{aligned} \tag{19.22}$$

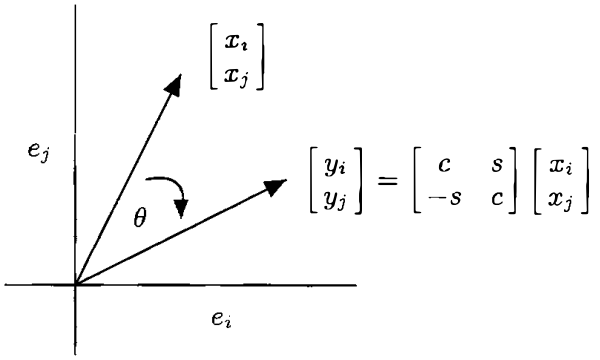

Figure 19.2. *Givens rotation,* $y = G(i,j,\theta)x$.

This result shows that the computed update is an exact orthogonal update of a perturbation of B, where the norm of the perturbation is bounded in terms of the error constants for the level-3 BLAS.

Two conclusions can be drawn. First, algorithms that employ the WY representation with conventional level-3 BLAS are as stable as the corresponding point algorithms. Second, the use of fast BLAS3 for applying the updates affects stability only through the constants in the backward error bounds. The same conclusions apply to the more storage-efficient compact WY representation of Schreiber and Van Loan [1024, 1989], and the variation of Puglisi [959, 1992].

19.6. Givens Rotations

Another way to compute the QR factorization is with Givens rotations. A Givens rotation (or plane rotation) $G(i,j,\theta) \in \mathbb{R}^{n\times n}$ is equal to the identity matrix except that

$$G([i,\,j],[i,\,j]) = \begin{bmatrix} c & s \\ -s & c \end{bmatrix},$$

where $c = \cos\theta$ and $s = \sin\theta$. The multiplication $y = G(i,j,\theta)x$ rotates x through θ radians clockwise in the (i,j) plane; see Figure 19.2. Algebraically,

$$y_k = \begin{cases} x_k, & k \neq i,j, \\ cx_i + sx_j, & k = i, \\ -sx_i + cx_j, & k = j, \end{cases}$$

and so $y_j = 0$ if

$$s = \frac{x_j}{\sqrt{x_i^2 + x_j^2}}, \quad c = \frac{x_i}{\sqrt{x_i^2 + x_j^2}}. \tag{19.23}$$

Givens rotations are therefore useful for introducing zeros into a vector one at a time. Note that there is no need to work out the angle θ, since c and s in (19.23) are all that are needed to apply the rotation. In practice, we would scale the computation to avoid overflow (cf. §27.8).

To compute the QR factorization, Givens rotations are used to eliminate the elements below the diagonal in a systematic fashion. Various choices and orderings of rotations can be used; a natural one is illustrated as follows for a generic 4×3 matrix:

$$A = \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} \xrightarrow{G_{34}} \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ 0 & \times & \times \end{bmatrix} \xrightarrow{G_{23}} \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix} \xrightarrow{G_{12}}$$

$$\left[\begin{array}{c|cc} \times & \times & \times \\ \hline 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{array}\right] \xrightarrow{G_{34}} \left[\begin{array}{c|cc} \times & \times & \times \\ \hline 0 & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \end{array}\right] \xrightarrow{G_{23}} \left[\begin{array}{cc|c} \times & \times & \times \\ 0 & \times & \times \\ \hline 0 & 0 & \times \\ 0 & 0 & \times \end{array}\right] \xrightarrow{G_{34}}$$

$$\begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & 0 \end{bmatrix} = R.$$

The operation count for Givens QR factorization of a general $m \times n$ matrix ($m \geq n$) is $3n^2(m - n/3)$ flops, which is 50% more than that for Householder QR factorization. The main use of Givens rotations is to operate on structured matrices—for example, to compute the QR factorization of a tridiagonal or Hessenberg matrix, or to carry out delicate zeroing in updating or downdating problems [509, 1996, §12.5].

Error analysis for Givens rotations is similar to that for Householder matrices—but a little easier. We omit the (straightforward) proof of the first result.

Lemma 19.7. *Let a Givens rotation $G(i,j,\theta)$ be constructed according to* (19.23). *The computed $\widehat{c}$ and $\widehat{s}$ satisfy*

$$\widehat{c} = c(1+\theta_4), \quad \widehat{s} = s(1+\theta_4'), \tag{19.24}$$

where $|\theta_4|, |\theta_4'| \leq \gamma_4$. □

Lemma 19.8. *Let $x \in \mathbb{R}^m$ and consider the computation of $y = \widehat{G}_{ij}x$, where $\widehat{G}_{ij}$ is a computed Givens rotation in the (i,j) plane for which $\widehat{c}$ and $\widehat{s}$ satisfy* (19.24). *The computed $\widehat{y}$ satisfies*

$$\widehat{y} = (G_{ij} + \Delta G_{ij})x, \qquad \|\Delta G_{ij}\|_F \leq \sqrt{2}\gamma_6,$$

where G_{ij} is an exact Givens rotation based on c and s in (19.24). *All the rows of ΔG_{ij} except the ith and jth are zero.*

Proof. The vector $\widehat{y}$ differs from x only in elements i and j. We have

$$\widehat{y}_i = fl(\widehat{c}x_i + \widehat{s}x_j) = cx_i(1+\theta_6) + sx_j(1+\theta_6'),$$

where $|\theta_6|, |\theta_6'| \leq \gamma_6$, and similarly for $\widehat{y}_j$. Hence

$$|\widehat{y} - G_{ij}x| \leq \gamma_6 |G_{ij}||x|,$$

so that $\|\widehat{y} - G_{ij}x\|_2 \le \sqrt{2}\gamma_6\|x\|_2$. We take $\Delta G_{ij} = (\widehat{y} - G_{ij}x)x^T/x^Tx$. □

For the next result we need the notion of disjoint Givens rotations. Rotations $G_{i_1,j_1}, \ldots, G_{i_r,j_r}$ are disjoint if the integers i_s, j_s, i_t, and j_t are distinct for $s \ne t$. Disjoint rotations are "nonconflicting" and therefore commute; it matters neither mathematically nor numerically in which order the rotations are applied. (Disjoint rotations can therefore be applied in parallel, though that is not our interest here.) Our approach is to take a given sequence of rotations and reorder them into groups of disjoint rotations. The reordered algorithm is numerically equivalent to the original one, but allows a simpler error analysis.

As an example of a rotation sequence already ordered into disjoint groups, consider the following sequence and ordering illustrated for a 6×5 matrix:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ 1 & \times & \times & \times & \times \\ 2 & 3 & \times & \times & \times \\ 3 & 4 & 5 & \times & \times \\ 4 & 5 & 6 & 7 & \times \\ 5 & 6 & 7 & 8 & 9 \end{bmatrix}.$$

Here, an integer k in position (i, j) denotes that the (i, j) element is eliminated on the kth step by a rotation in the (j, i) plane, and all rotations on the kth step are disjoint. For an $m \times n$ matrix with $m > n$ there are $r = m + n - 2$ stages, and the Givens QR factorization can be written as $W_rW_{r-1}\ldots W_1A = R$, where each W_i is a product of at most n disjoint rotations. It is easy to see that an analogous grouping into disjoint rotations can be done for the scheme illustrated at the start of this section.

Lemma 19.9. *Consider the sequence of transformations*

$$A_{k+1} = W_kA_k, \qquad k = 1\colon r,$$

where $A_1 = A \in \mathbb{R}^{m\times n}$ and each W_k is a product of disjoint Givens rotations. Assume that the individual Givens rotations are performed using computed sine and cosine values related to the exact values defining the W_k by (19.24). *Then the computed matrix $\widehat{A}_{r+1}$ satisfies*

$$\widehat{A}_{r+1} = Q^T(A + \Delta A),$$

where $Q^T = W_rW_{r-1}\ldots W_1$ and

$$\|\Delta a_j\|_2 \le \tilde{\gamma}_r\|a_j\|_2, \qquad j = 1\colon n.$$

In the special case $n = 1$, so that $A = a$, we have $\widehat{a}^{(r+1)} = (Q + \Delta Q)^Ta$ with $\|\Delta Q\|_F \le \tilde{\gamma}_r$.

Proof. The proof is analogous to that of Lemma 19.3, so we offer only a sketch. First, we consider the jth column of A, a_j, which undergoes the transformations $a_j^{(r+1)} = W_r\ldots W_1a_j$. By Lemma 19.8 and the disjointness of the rotations, we have

$$\widehat{a}_j^{(r+1)} = (W_r + \Delta W_r)\ldots(W_1 + \Delta W_1)a_j,$$

where each ΔW_k depends on j and satisfies $\|\Delta W_k\|_2 \le \sqrt{2}\gamma_6$. Using Lemma 3.6 we obtain

$$\begin{aligned} \widehat{a}_j^{(r+1)} &= Q^T(a_j + \Delta a_j), \\ \|\Delta a_j\|_2 &\le \bigl((1+\sqrt{2}\gamma_6)^r - 1\bigr)\|a_j\|_2 = \tilde{\gamma}_r\|a_j\|_2. \end{aligned} \tag{19.25}$$

The result for $n = 1$ is proved as in Lemma 19.3. □

We are now suitably equipped to give a result for Givens QR factorization.

Theorem 19.10. *Let $\widehat{R} \in \mathbb{R}^{m\times n}$ be the computed upper trapezoidal QR factor of $A \in \mathbb{R}^{m\times n}$ $(m \ge n)$ obtained via the Givens QR algorithm, with any standard choice and ordering of rotations. Then there exists an orthogonal $Q \in \mathbb{R}^{m\times m}$ such that*

$$A + \Delta A = Q\widehat{R}, \qquad \|\Delta a_j\|_2 \le \tilde{\gamma}_{m+n-2}\|a_j\|_2, \quad j = 1\colon n.$$

(The matrix Q is a product of Givens rotations, the kth of which corresponds to the exact application of the kth step of the algorithm to $\widehat{A}_k$.) □

It is interesting that the error bounds for QR factorization with Givens rotations are a factor n smaller than those for Householder QR factorization. This appears to be an artefact of the analysis, and we are not aware of any difference in accuracy in practice.

19.7. Iterative Refinement

Consider a nonsingular linear system $Ax = b$, where $A \in \mathbb{R}^{n\times n}$. Suppose we solve the system using a QR factorization $A = QR$ computed using Householder or Givens transformations (thus, x is obtained by solving $Rx = Q^Tb$). Theorem 19.5, and its obvious analogue for Givens rotations, imply a small columnwise relative backward error but not a small componentwise relative backward error. In fact, we know of no nontrivial class of matrices for which Householder or Givens QR factorization is guaranteed to yield a small componentwise relative backward error.

Suppose that we carry out a step of fixed precision iterative refinement, to obtain $\widehat{y}$. In order to invoke Theorem 12.4 we need to express the backward error bounds for $\widehat{x}$ in the form

$$|b - A\widehat{x}| \le u\bigl(G|A||\widehat{x}| + H|b|\bigr),$$

for suitable nonnegative matrices G and H. For our columnwise backward error bounds this can be done with the aid of Lemma 6.6: (19.14) yields

$$|b - A\widehat{x}| \le |\Delta A||\widehat{x}| \le \tilde{\gamma}_{n^2}\, ee^T|A||\widehat{x}|.$$

Theorem 12.4 can now be invoked with $G \approx n^2ee^T$ and $H = 0$, giving the conclusion that the componentwise relative backward error $\omega_{|A|,|b|}(\widehat{y})$ after one step of iterative refinement will be small provided that A is not too ill conditioned and $|A||\widehat{y}|$ is not too badly scaled. This conclusion is similar to that for GEPP, except that for GEPP there is the added requirement that the LU factorization does not suffer large element growth.

The limiting forward error can be bounded by Theorems 12.1 and 12.2, for which $\eta \approx n^2 u \kappa_\infty(A)$.

The performance of QR factorization with fixed precision iterative refinement is illustrated in Tables 12.1–12.3 in §12.2. The performance is as predicted by the analysis. Notice that the initial componentwise relative backward error is large in Table 12.2 but that iterative refinement successfully reduces it to the roundoff level (despite $\mathrm{cond}(A^{-1})\sigma(A,x)$ being huge). It is worth stressing that the QR factorization yielded a small *normwise* relative backward error in each example ($\eta_{A,b}(\widehat{x}) < u$, in fact), as we know it must.

19.8. Gram–Schmidt Orthogonalization

The oldest method for computing a QR factorization is the Gram–Schmidt orthogonalization method. It can be derived directly from the equation $A = QR$, where $A, Q \in \mathbb{R}^{m\times n}$ and $R \in \mathbb{R}^{n\times n}$ (Gram–Schmidt does not compute the $m \times m$ matrix Q in the full QR factorization and hence does not provide a basis for the orthogonal complement of range(A).) Denoting by a_j and q_j the jth columns of A and Q, respectively, we have

$$a_j = \sum_{k=1}^{j} r_{kj} q_k.$$

Premultiplying by q_i^T yields, since Q has orthonormal columns, $q_i^T a_j = r_{ij}$, $i = 1\colon j-1$. Further,

$$q_j = q_j'/r_{jj},$$

where

$$q_j' = a_j - \sum_{k=1}^{j-1} r_{kj} q_k, \quad r_{jj} = \|q_j'\|_2.$$

Hence we can compute Q and R a column at a time. To ensure that $r_{jj} > 0$ we require that A has full rank.

Algorithm 19.11 (classical Gram–Schmidt). Given $A \in \mathbb{R}^{m\times n}$ of rank n this algorithm computes the QR factorization $A = QR$, where Q is $m \times n$ and R is $n \times n$, by the Gram–Schmidt method.

```
for j = 1:n
    for i = 1:j-1
        r_ij = q_i^T a_j
    end
    q_j' = a_j - sum_{k=1}^{j-1} r_kj q_k
    r_jj = ||q_j'||_2
    q_j = q_j'/r_jj
end
```

Cost: $2mn^2$ flops ($2n^3/3$ flops more than Householder QR factorization with Q left in factored form).

In the classical Gram–Schmidt method (CGS), a_j appears in the computation only at the jth stage. The method can be rearranged so that as soon as q_j is computed, all the remaining vectors are orthogonalized against q_j. This gives the modified Gram–Schmidt method (MGS).

Algorithm 19.12 (modified Gram–Schmidt). Given $A \in \mathbb{R}^{m\times n}$ of rank n this algorithm computes the QR factorization $A = QR$, where Q is $m \times n$ and R is $n \times n$, by the MGS method.

$a_k^{(1)} = a_k$, $k = 1\colon n$
for $k = 1\colon n$
  $r_{kk} = \|a_k^{(k)}\|_2$
  $q_k = a_k^{(k)}/r_{kk}$
  for $j = k+1\colon n$
    $r_{kj} = q_k^T a_j^{(k)}$
    $a_j^{(k+1)} = a_j^{(k)} - r_{kj}q_k$
  end
end

Cost: $2mn^2$ flops.

It is worth noting that there are two differences between the CGS and MGS methods. The first is the order in which the calculations are performed: in the modified method each remaining vector is updated once on each step instead of having all its updates done together on one step. This is purely a matter of the order in which the operations are performed. Second, and more crucially in finite precision computation, two different (but mathematically equivalent) formulae for r_{kj} are used: in the classical method, $r_{kj} = q_k^T a_j$, which involves the original vector a_j, whereas in the modified method a_j is replaced in this formula by the partially orthogonalized vector $a_j^{(k)}$. Another way to view the difference between the two Gram–Schmidt methods is via representations of an orthogonal projection; see Problem 19.8.

The MGS procedure can be expressed in matrix terms by defining $A_k = [q_1, \dots, q_{k-1}, a_k^{(k)}, \dots, a_n^{(k)}]$. MGS transforms $A_1 = A$ into $A_{n+1} = Q$ by the sequence of transformations $A_k = A_{k+1}R_k$, where R_k is equal to the identity except in the kth row, where it agrees with the final R. For example, if $n = 4$ and $k = 3$,

$$A_3 = \begin{bmatrix} q_1 & q_2 & a_3^{(3)} & a_4^{(3)} \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & q_3 & a_4^{(4)} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & r_{33} & r_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_4 R_3.$$

Thus $R = R_n \dots R_1$.

The Gram–Schmidt methods produce Q explicitly, unlike the Householder and Givens methods, which hold Q in factored form. While this is a benefit, in that no extra work is required to form Q, it is also a weakness, because there is nothing in the methods to force the computed $\widehat{Q}$ to be orthonormal in the face of roundoff.

Orthonormality of Q is a consequence of orthogonality relations that are implicit in the methods, and these relations may be vitiated by rounding errors.

Some insight is provided by the case $n = 2$, for which the CGS and MGS methods are identical. Given $a_1, a_2 \in \mathbb{R}^m$ we compute $q_1 = a_1/\|a_1\|_2$, which we will suppose is done exactly, and then we form the unnormalized vector $q_2 = a_2 - (q_1^T a_2)q_1$. The computed vector satisfies

$$\widehat{q}_2 = a_2 - q_1^T(a_2 + \Delta a_2)q_1 + \Delta\widetilde{q}_2,$$

where

$$|\Delta a_2| \le \gamma_m |a_2|, \quad |\Delta\widetilde{q}_2| \le u|a_2| + \gamma_2 |q_1^T(a_2 + \Delta a_2)||q_1|.$$

Hence

$$\widehat{q}_2 = q_2 + \Delta q_2, \quad |\Delta q_2| \le \gamma_m |q_1^T||a_2||q_1| + u|a_2| + \gamma_2(1+\gamma_m)|q_1^T||a_2||q_1|,$$

and so the normalized inner product satisfies

$$\left| q_1^T \frac{\widehat{q}_2}{\|\widehat{q}_2\|_2} \right| \lesssim (m+2)u \frac{\|a_2\|_2}{\|q_2\|_2} = \frac{(m+2)u}{\sin\angle(a_1, a_2)}, \tag{19.26}$$

where $\angle(a_1, a_2)$ is the angle between a_1 and a_2. But $\kappa_2(A) \ge \cot\angle(a_1, a_2)$, where $A = [a_1,\ a_2]$ (Problem 19.9). Hence, for $n = 2$, the loss of orthogonality can be bounded in terms of $\kappa_2(A)$. The same is true in general for the MGS method, as proved by Björck [119, 1967]. A direct proof is quite long and complicated, but a recent approach of Björck and Paige [131, 1992] enables a much shorter derivation; we take this approach here.

The observation that simplifies the error analysis of the MGS method is that the method is equivalent, both mathematically *and numerically*, to Householder QR factorization of the padded matrix $\begin{bmatrix} 0_n \\ A \end{bmatrix} \in \mathbb{R}^{(m+n)\times n}$. To understand this equivalence, consider the Householder QR factorization

$$P^T \begin{bmatrix} 0_n \\ A \end{bmatrix} = \begin{bmatrix} R \\ 0 \end{bmatrix}, \qquad P^T = P_n \dots P_2 P_1. \tag{19.27}$$

Let $q_1, \dots, q_n \in \mathbb{R}^m$ be the vectors obtained by applying the MGS method to A. Then it is easy to see that

$$P_1 = I - v_1 v_1^T, \quad v_1 = \begin{bmatrix} -e_1 \\ q_1 \end{bmatrix}, \quad v_1^T v_1 = 2$$

and that the multiplication $A_2 = P_1 \begin{bmatrix} 0_n \\ A \end{bmatrix}$ carries out the first step of the MGS method on A, producing the first row of R and $a_2^{(2)}, \dots, a_n^{(2)}$:

$$A_2 = \begin{matrix} 1 \\ n-1 \\ m \end{matrix} \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ 0 & 0 & \dots & 0 \\ 0 & a_2^{(2)} & \dots & a_n^{(2)} \end{bmatrix}.$$

The argument continues in the same way, and we find that

$$P_k = I - v_k v_k^T, \quad v_k = \begin{bmatrix} -e_k \\ q_k \end{bmatrix}, \quad v_k^T v_k = 2, \quad k = 2\colon n. \tag{19.28}$$

With the Householder–MGS connection established, we are ready to derive error bounds for the MGS method by making use of our existing error analysis for the Householder method.

Theorem 19.13. *Suppose the MGS method is applied to $A \in \mathbb{R}^{m\times n}$ of rank n, yielding computed matrices $\widehat{Q} \in \mathbb{R}^{m\times n}$ and $\widehat{R} \in \mathbb{R}^{n\times n}$. Then there are constants $c_i \equiv c_i(m,n)$ such that*

$$A + \Delta A_1 = \widehat{Q}\widehat{R}, \qquad \|\Delta A_1\|_2 \le c_1 u\|A\|_2, \tag{19.29}$$

$$\|\widehat{Q}^T\widehat{Q} - I\|_2 \le c_2 u\kappa_2(A) + O\big((u\kappa_2(A))^2\big), \tag{19.30}$$

and there exists an orthonormal matrix Q such that

$$A + \Delta A_2 = Q\widehat{R}, \qquad \|\Delta A_2(:,j)\|_2 \le c_3 u\|a_j\|_2, \quad j = 1\colon n. \tag{19.31}$$

Proof. To prove (19.29) we use the matrix form of the MGS method. For the computed matrices we have

$$\widehat{A}_k = \widehat{A}_{k+1}\widehat{R}_k + \Delta_k, \qquad |\Delta_k| \le u|\widehat{A}_{k+1}||\widehat{R}_k|.$$

Expanding this recurrence, we obtain

$$A = \widehat{Q}\widehat{R} + \Delta_n\widehat{R}_{n-1}\dots\widehat{R}_1 + \Delta_{n-1}\widehat{R}_{n-2}\dots\widehat{R}_1 + \cdots + \Delta_2\widehat{R}_1 + \Delta_1.$$

Hence

$$|A - \widehat{Q}\widehat{R}| \le u\big(|\widehat{A}_{n+1}||\widehat{R}_n|\dots|\widehat{R}_1| + \cdots + |\widehat{A}_3||\widehat{R}_2||\widehat{R}_1| + |\widehat{A}_2||\widehat{R}_1|\big), \tag{19.32}$$

and a typical term has the form

$$|\widehat{A}_k||\widehat{R}_{k-1}|\dots|\widehat{R}_1| = \big|[\widehat{q}_1 \ \dots \ \widehat{q}_{k-1} \ \widehat{a}_k^{(k)} \ \dots \ \widehat{a}_n^{(k)}]\big|S_{k-1}, \tag{19.33}$$

where S_{k-1} agrees with $|\widehat{R}|$ in its first $k-1$ rows and the identity in its last $n-k+1$ rows. Assume for simplicity that $\|\widehat{q}_i\|_2 \equiv 1$ (this does not affect the final result). We have $a_j^{(k+1)} = (I - q_k q_k^T)a_j^{(k)}$, and Lemma 3.9 shows that the computed vector satisfies

$$\|\widehat{a}_j^{(k+1)}\|_2 \le (1 + 2\gamma_{m+3})\|\widehat{a}_j^{(k)}\|_2,$$

which implies $\|\widehat{A}_{k+1}\|_F \le (1+2\gamma_{m+3})^k\|A\|_F$ and, from $\widehat{r}_{kj} = fl(\widehat{q}_k^T\widehat{a}_j^{(k)})$, we have $\|\widehat{R}\|_F \le \sqrt{n}(1+\gamma_m)(1+2\gamma_{m+3})^{n-1}\|A\|_F$. Using (19.32) and exploiting the form of (19.33) we find, after a little working, that

$$\|A - \widehat{Q}\widehat{R}\|_F \le 4n^2 u\|A\|_F,$$

provided that $(1+\gamma_m)(1+2\gamma_{m+3})^{n-1} < 2$.

To prove the last two parts of the theorem we exploit the Householder–MGS connection. By applying Theorem 19.4 to (19.27) we find that there is an orthogonal $\widetilde{P}$ such that

$$\begin{bmatrix} \Delta A_3 \\ A + \Delta A_4 \end{bmatrix} = \widetilde{P}\begin{bmatrix} \widehat{R} \\ 0 \end{bmatrix} = \begin{bmatrix} \widetilde{P}_{11} \\ \widetilde{P}_{21} \end{bmatrix}\widehat{R}, \tag{19.34}$$

with

$$\left\| \begin{bmatrix} \Delta A_3(:,j) \\ \Delta A_4(:,j) \end{bmatrix} \right\|_2 \le \tilde{\gamma}_{mn} \|a_j\|_2, \qquad j = 1\colon n.$$

This does not directly yield (19.31), since $\widetilde{P}_{21}$ is not orthonormal. However, it can be shown that if we define Q to be the nearest orthonormal matrix to $\widetilde{P}_{21}$ in the Frobenius norm, then (19.31) holds with $c_3 = (\sqrt{m}+1)c_4$ (see Problem 19.12).

Now (19.29) and (19.31) yield $\widehat{Q} - Q = (\Delta A_1 - \Delta A_2)\widehat{R}^{-1}$, so

$$\|\widehat{Q} - Q\|_2 \le (c_1 + \sqrt{n}c_3)u\|A\|_2\|\widehat{R}^{-1}\|_2 \le \frac{c_5 u\kappa_2(A)}{1 - \sqrt{n}c_3 u\kappa_2(A)},$$

where $c_5 = c_1 + \sqrt{n}c_3$ and we have used (19.31) to bound $\|\widehat{R}^{-1}\|_2$. This bound implies (19.30) with $c_2 = 2c_5$ (use the first inequality in Problem 19.14). □

We note that (19.30) can be strengthened by replacing $\kappa_2(A)$ in the bound by the minimum over positive diagonal matrices D of $\kappa_2(AD)$. This follows from the observation that in the MGS method the computed $\widehat{Q}$ is invariant under scalings $A \leftarrow AD$, at least if D comprises powers of the machine base. As a check, note that the bound in (19.26) for the case $n = 2$ is independent of the column scaling, since $\sin \angle(a_1, a_2)$ is.

Theorem 19.13 tells us three things. First, the computed QR factors from the MGS method have a small residual. Second, the departure from orthonormality of $\widehat{Q}$ is bounded by a multiple of $\kappa_2(A)u$, so that $\widehat{Q}$ is guaranteed to be nearly orthonormal if A is well conditioned. Finally, $\widehat{R}$ is the exact triangular QR factor of a matrix near to A in a columnwise sense, so it is as good an R-factor as that produced by Householder QR factorization applied to A. In terms of the error analysis, the MGS method is weaker than Householder QR factorization *only* in that $\widehat{Q}$ is not guaranteed to be nearly orthonormal.

For the CGS method the residual bound (19.29) still holds, but no bound of the form (19.30) holds for $n > 2$ (see Problem 19.10).

Here is a numerical example to illustrate the behaviour of the Gram–Schmidt methods. We take the 25×15 Vandermonde matrix $A = (p_i^{j-1})$, where the p_i are equally spaced on $[0,1]$. The condition number $\kappa_2(A) = 1.5 \times 10^9$. We have

$$\begin{aligned} \text{CGS:} \quad & \|A - \widehat{Q}\widehat{R}\|_2 = 5.0 \times 10^{-16}, \quad \|\widehat{Q}^T\widehat{Q} - I\|_2 = 5.2, \\ \text{MGS:} \quad & \|A - \widehat{Q}\widehat{R}\|_2 = 1.0 \times 10^{-15}, \quad \|\widehat{Q}^T\widehat{Q} - I\|_2 = 9.5 \times 10^{-9}. \end{aligned}$$

Both methods produce a small residual for the QR factorization. While CGS produces a $\widehat{Q}$ showing no semblance of orthogonality, for MGS we have $\|\widehat{Q}^T\widehat{Q} - I\|_2 \approx \kappa_2(A)u/17$.

19.9. Sensitivity of the QR Factorization

How do the QR factors of a matrix behave under small perturbations of the matrix? This question was first considered by Stewart [1068, 1977]. He showed that if $A \in \mathbb{R}^{m\times n}$ has rank n and

$$A = QR \quad \text{and} \quad A + \Delta A = (Q + \Delta Q)(R + \Delta R)$$

are QR factorizations, then, for sufficiently small ΔA,

$$\frac{\|\Delta R\|_F}{\|R\|_F} \leq c_n \kappa_F(A) \frac{\|\Delta A\|_F}{\|A\|_F}, \tag{19.35a}$$

$$\|\Delta Q\|_F \leq c_n \kappa_F(A) \frac{\|\Delta A\|_F}{\|A\|_F}, \tag{19.35b}$$

where c_n is a constant. Here, and throughout this section, we use the "economy size" QR factorization with R a square matrix normalized to have nonnegative diagonal elements. Similar normwise bounds are given by Stewart [1075, 1993] and Sun [1102, 1991], and, for ΔQ only, by Bhatia and Mukherjea [108, 1994] and Sun [1106, 1995].

Columnwise sensitivity analyses have been given by Zha [1278, 1993] and Sun [1103, 1992], [1104, 1992]. Zha's bounds can be summarized as follows, with the same assumptions and notation as for Stewart's result above. Let $|\Delta A| \leq \epsilon G|A|$, where G is nonnegative with $\|G\|_2 = 1$. Then, for sufficiently small ϵ,

$$\begin{aligned} \frac{\|\Delta R\|_2}{\|R\|_2} &\leq c_{m,n}\epsilon \,\mathrm{cond}(R^{-1}) + O(\epsilon^2), \\ \|\Delta Q\|_2 &\leq c_{m,n}\epsilon \,\mathrm{cond}(R^{-1}) + O(\epsilon^2), \end{aligned} \tag{19.36}$$

where $c_{m,n}$ is a constant depending on m and n. The quantity $\phi(A) = \mathrm{cond}(R^{-1}) = \| \, |R||R^{-1}| \, \|_2$ can therefore be thought of as a condition number for the QR factorization under the columnwise class of perturbations considered. Note that ϕ is independent of the column scaling of A.

As an application of these bounds, consider a computed QR factorization $A \approx \widehat{Q}\widehat{R}$ obtained via the Householder algorithm, where $\widehat{Q}$ is the computed product of the computed Householder matrices, and let Q be the exact Q-factor of A. Theorem 19.4 shows that $A + \Delta A = \widetilde{Q}\widehat{R}$ for an exactly orthogonal $\widetilde{Q}$, with $|\Delta A| \leq \tilde{\gamma}_{mn} ee^T |A|$ (cf. §19.7). Moreover, we know from (19.13) that $\|\widehat{Q} - \widetilde{Q}\|_F \leq \sqrt{n}\tilde{\gamma}_{mn}$. Now $Q - \widehat{Q} = (Q - \widetilde{Q}) + (\widetilde{Q} - \widehat{Q})$ and hence applying (19.36) to the first term we obtain

$$\|Q - \widehat{Q}\|_2 \leq c_{m,n} u \phi(A) + O(u^2). \tag{19.37}$$

In cases where a large $\kappa_F(A)$ is caused by poor column scaling, we can improve the bounds (19.35) by undoing the poor scaling to leave a well-conditioned matrix; the virtue of the columnwise analysis is that it does not require a judicious scaling in order to yield useful results.

19.10. Notes and References

The earliest appearance of Householder matrices is in the book by Turnbull and Aitken [1168, 1932, pp. 102–105]. These authors show that if $\|x\|_2 = \|y\|_2$ ($x \neq -y$) then a unitary matrix of the form $R = \alpha zz^* - I$ (in their notation) can be constructed so that $Rx = y$. They use this result to prove the existence of the Schur decomposition. The first systematic use of Householder matrices for computational purposes was by Householder [643, 1958], who used them to construct the QR factorization. Householder's motivation was to compute the QR

factorization with fewer arithmetic operations (in particular, fewer square roots) than are required by the use of Givens rotations.

A detailed analysis of different algorithms for constructing a Householder matrix P such that $Px = \sigma e_1$ is given by Parlett [924, 1971].

Tsao [1161, 1975] describes an alternative way to form the product of a Householder matrix with a vector and gives an error analysis. There is no major advantage over the usual approach.

As for Householder matrices, normwise error analysis for Givens rotations was given by Wilkinson [1231, 1963], [1233, 1965, pp. 131–139]. Wilkinson analysed QR factorization by Givens rotations for square matrices [1233, 1965, pp. 240–241], and his analysis was extended to rectangular matrices by Gentleman [474, 1973]. The idea of exploiting disjoint rotations in the error analysis was developed by Gentleman [475, 1975], who gave a normwise analysis that is simpler and produces smaller bounds than Wilkinson's (our normwise bound in Theorem 19.10 is essentially the same as Gentleman's).

For more details of algorithmic and other aspects of Householder and Givens QR factorization, see Golub and Van Loan [509, 1996, §5].

The error analysis in §19.3 is a refined and improved version of analysis that appeared in the technical report [594, 1990] and was quoted without proof in Higham [596, 1991]. The analysis has been reworked for this edition of the book to emphasize the columnwise nature of the backward error bounds.

The need for row and column pivoting in Householder QR factorization for badly row scaled matrices was established by Powell and Reid [951, 1969] and was reported in Lawson and Hanson's 1974 book [775, pp. 103–106, 149]. Theorem 19.6 was originally proved under some additional assumptions by Powell and Reid [951, 1969]. The result as stated is proved by Cox and Higham [275, 1998]; it also follows from a more general result of Higham [614, 2000] that includes Theorems 19.4 and 19.6 as special cases. Björck [128, 1996, p. 169] conjectures that "there is no need to perform row pivoting in Householder QR, provided that the rows are sorted after decreasing row norm before the factorization". This conjecture was proved by Cox and Higham [275, 1998], who also pointed out that row-wise backward stability is obtained for only one of the two possible choices of sign in the Householder vector.

The WY representation for a product of Householder transformations should not be confused with a genuine block Householder transformation. Schreiber and Parlett [1023, 1988] define, for a given $Z \in \mathbb{R}^{m\times n}$ $(m \geq n)$, the "block reflector that reverses the range of Z" as

$$H = I_m - ZWZ^T, \qquad W = 2(Z^TZ)^+ \in \mathbb{R}^{n\times n}.$$

If $n = 1$ this is just a standard Householder transformation. A basic task is as follows: given $E \in \mathbb{R}^{m\times n}$ $(m > n)$ find a block reflector H such that

$$HE = \begin{bmatrix} F \\ 0 \end{bmatrix}, \qquad F \in \mathbb{R}^{n\times n}.$$

Schreiber and Parlett develop theory and algorithms for block reflectors, in both of which the polar decomposition plays a key role.

Sun and Bischof [1111, 1995] show that any orthogonal matrix can be expressed in the form $Q = I - YSY^T$, even with S triangular, and they explore the properties of this representation.

Another important use of Householder matrices, besides computation of the QR factorization, is to reduce a matrix to a simpler form prior to iterative computation of eigenvalues (Hessenberg or tridiagonal form) or singular values (bidiagonal form). For these two-sided transformations an analogue of Lemma 19.3 holds with normwise bounds (only) on the perturbation. Error analyses of two-sided application of Householder transformations is given by Ortega [907, 1963] and Wilkinson [1230, 1962], [1233, 1965, Chap. 6].

Mixed precision iterative refinement for solution of linear systems by Householder QR factorization is discussed by Wilkinson [1234, 1965, §10], who notes that convergence is obtained as long as a condition of the form $c_n \kappa_2(A) u < 1$ holds.

Fast Givens rotations can be applied to a matrix with half the number of multiplications of conventional Givens rotations, and they do not involve square roots. They were developed by Gentleman [474, 1973] and Hammarling [542, 1974]. Fast Givens rotations are as stable as conventional ones—see the error analysis by Parlett in [926, 1998, §6.8.3], for example—but, for the original formulations, careful monitoring is required to avoid overflow. Rath [973, 1982] investigates the use of fast Givens rotations for performing similarity transformations in solving the eigenproblem. Barlow and Ipsen [74, 1987] propose a class of scaled Givens rotations suitable for implementation on systolic arrays, and they give a detailed error analysis. Anda and Park [19, 1994] develop fast rotation algorithms that use dynamic scaling to avoid overflow.

Rice [984, 1966] was the first to point out that the MGS method produces a more nearly orthonormal matrix than the CGS method in the presence of rounding errors. Björck [119, 1967] gives a detailed error analysis, proving (19.29) and (19.30) but not (19.31), which is an extension of the corresponding normwise result of Björck and Paige [131, 1992]. Björck and Paige give a detailed assessment of MGS versus Householder QR factorization.

The difference between the CGS and MGS methods is indeed subtle. Wilkinson [1239, 1971] admitted that "I used the modified process for many years without even noticing explicitly that I was not performing the classical algorithm."

The orthonormality of the matrix $\widehat{Q}$ from Gram–Schmidt can be improved by reorthogonalization, in which the orthogonalization step of the classical or modified method is iterated. Analyses of Gram–Schmidt with reorthogonalization are given by Abdelmalek [2, 1971], Ruhe [996, 1983], and Hoffmann [634, 1989]. Daniel, Gragg, Kaufman, and Stewart [290, 1976] analyse the use of classical Gram–Schmidt with reorthogonalization for updating a QR factorization after a rank one change to the matrix.

The mathematical and numerical equivalence of the MGS method with Householder QR factorization of the matrix $\left[\begin{smallmatrix} 0_n \\ A \end{smallmatrix}\right]$ was known in the 1960s (see the Björck quotation at the start of the chapter) and the mathematical equivalence was pointed out by Lawson and Hanson [775, 1995, Ex. 19.39].

A block Gram–Schmidt method is developed by Jalby and Philippe [668, 1991] and error analysis given. See also Björck [127, 1994], who gives an up-to-date

survey of numerical aspects of the Gram–Schmidt method.

For more on Gram–Schmidt methods, including historical comments, see Björck [128, 1996].

More refined (and substantially more complicated) perturbation bounds for the QR factorization than those in §19.9 are given by Chang, Paige and Stewart [221, 1997] and Chang and Paige [219, 2001].

One use of the QR factorization is to orthogonalize a matrix that, because of rounding or truncation errors, has lost its orthogonality; thus we compute $A = QR$ and replace A by Q. An alternative approach is to replace $A \in \mathbb{R}^{m\times n}$ ($m \geq n$) by the nearest orthonormal matrix, that is, the matrix Q that solves $\{\|A - Q\| : Q^TQ = I\} = \min$. For the 2- and Frobenius norms, the optimal Q is the orthonormal polar factor U of A, where $A = UH$ is a *polar decomposition*: $U \in \mathbb{R}^{m\times n}$ has orthonormal columns and $H \in \mathbb{R}^{n\times n}$ is symmetric positive semidefinite. If $m = n$, U is the nearest orthogonal matrix to A in any unitarily invariant norm, as shown by Fan and Hoffman [401, 1955]. Chandrasekaran and Ipsen [216, 1994] show that the QR and polar factors satisfy $\|A - Q\|_{2,F} \leq 5\sqrt{n}\|A - U\|_2$, under the assumptions that A has full rank and columns of unit 2-norm and that R has positive diagonal elements. Sun [1105, 1995] proves a similar result and also obtains a bound for $\|Q-U\|_F$ in terms of $\|A^TA-I\|_F$. Algorithms for maintaining orthogonality in long products of orthogonal matrices, which arise, for example, in subspace tracking problems in signal processing, are analysed by Edelman and Stewart [384, 1993] and Mathias [823, 1996].

Various iterative methods are available for computing the orthonormal polar factor U, and they can be competitive in cost with computation of a QR factorization. For more details on the theory and numerical methods, see Higham [578, 1986], [587, 1989], Higham and Papadimitriou [621, 1994], and the references therein.

A notable omission from this chapter is a treatment of rank-revealing QR factorizations—ones in which the rank of A can readily be determined from R. This topic is not one where rounding errors play a major role, and hence it is outside the scope of this book. Pointers to the literature include Golub and Van Loan [509, 1996, §5.4], Chan and Hansen [212, 1992], and Björck [128, 1996]. Column pivoting in the QR factorization ensures that if A has rank r then only the first r rows of R are nonzero (see Problem 19.5). A perturbation theorem for the QR factorization with column pivoting is given by Higham [588, 1990]; it is closely related to the perturbation theory in §10.3.1 for the Cholesky factorization of a positive semidefinite matrix.

19.10.1. LAPACK

LAPACK contains a rich selection of routines for computing and manipulating the QR factorization and its variants. Routine `xGEQRF` computes the QR factorization $A = QR$ of an $m \times n$ matrix A by the Householder QR algorithm. If $m < n$ (which we ruled out in our analysis, merely to simplify the notation), the factorization takes the form $A = Q[R_1\ R_2]$, where R_1 is $m \times m$ upper triangular. The matrix Q is represented as a product of Householder transformations and is not formed explicitly. A routine `xORGQR` (or `xUNGQR` in the complex case) is provided to form

all or part of Q, and routine `xORMQR` (or `xUNMQR`) will pre- or postmultiply a matrix by Q or its (conjugate) transpose.

Routine `xGEQPF` computes the QR factorization with column pivoting.

An LQ factorization is computed by `xGELQF`. When A is $m \times n$ with $m \le n$ it takes the form $A = [L \;\; 0]\,Q$. It is essentially the same as a QR factorization of A^T and hence can be used to find the minimum 2-norm solution to an underdetermined system (see §21.1).

LAPACK also computes two nonstandard factorizations of an $m \times n$ A:

$$\texttt{xGEQLF}: \; A = Q \begin{bmatrix} 0 \\ L \end{bmatrix}, \quad m \ge n, \qquad \texttt{xGERQF}: \; A = [0 \quad R]\,Q, \quad m \le n,$$

where L is lower trapezoidal and R upper trapezoidal.

Problems

19.1. Find the eigenvalues of a Householder matrix and a Givens matrix.

19.2. Let $\widehat{P} = I - \widehat{\beta}\widehat{v}\widehat{v}^T$, where $\widehat{\beta}$ and $\widehat{v}$ are the computed quantities described in Lemma 19.1. Derive a bound for $\|\widehat{P}^T\widehat{P} - I\|_2$.

19.3. A complex Householder matrix has the form

$$P = I - \beta v v^*,$$

where $0 \ne v \in \mathbb{C}^n$ and $\beta = 2/v^*v$. For given $x, y \in \mathbb{C}^n$, show how to determine, if possible, P so that $Px = y$.

19.4. (Wilkinson [1233, 1965, p. 242]) Let $x \in \mathbb{R}^n$ and let P be a Householder matrix such that $Px = \pm\|x\|_2 e_1$. Let $G_{1,2}, \ldots, G_{n-1,n}$ be Givens rotations such that $Qx := G_{1,2} \ldots G_{n-1,n} x = \pm\|x\|_2 e_1$. True or false: $P = Q$?

19.5. Show that the R factor produced by QR factorization with column pivoting (see (19.15)) satisfies

$$r_{kk}^2 \ge \sum_{i=k}^{j} r_{ij}^2, \qquad j = k+1\colon n, \quad k = 1\colon n,$$

so that, in particular, $|r_{11}| \ge |r_{22}| \ge \cdots \ge |r_{nn}|$. (These are the same equations as (10.13), which hold for the Cholesky factorization with complete pivoting—why?)

19.6. (Higham [614, 2000]) Show that in Householder QR factorization applied to $A \in \mathbb{R}^{m\times n}$ the Householder vector v_k from the kth stage constructed according to (19.1) satisfies

$$\sqrt{2}\|a_k^{(k)}(k\colon m)\|_2 \le \|v_k\|_2 \le 2\|a_k^{(k)}(k\colon m)\|_2. \tag{19.38}$$

Consider now the computation of $\widehat{a}_j^{(k+1)} = fl(\widehat{P}_k \widehat{a}_j^{(k)})$ for $j > k$, where $\widehat{P}_k = I - \beta_k \widehat{v}_k \widehat{v}_k^T$ and $\widehat{v}_k$ satisfies

$$\widehat{v}_k = v_k + \Delta v_k, \qquad |\Delta v_k| \le \tilde{\gamma}_{m-k}|v_k|,$$

where

$$P_k = I - \beta_k v_k v_k^T$$

is the Householder matrix corresponding to the exact application of the kth stage of the algorithm to the computed matrix $\widehat{A}^{(k)}$. Show that

$$\widehat{a}_j^{(k+1)} = P_k \widehat{a}_j^{(k)} + f_j^{(k)}, \tag{19.39}$$

where $f_j^{(k)}(1\colon k-1) = 0$ and

$$|f_j^{(k)}| \le u|\widehat{a}_j^{(k)}| + \tilde{\gamma}_{m-k} \frac{\||\widehat{a}_j^{(k)}|(k\colon m)\|_2}{\|\widehat{a}_k^{(k)}(k\colon m)\|_2} |v_k|. \tag{19.40}$$

Explain the significance of this result for badly row-scaled problems.

19.7. Let $W \in \mathbb{R}^{m\times m}$ be a product of disjoint Givens rotations. Show that $\| |W| \|_2 \le \sqrt{2}$.

19.8. The CGS method and the MGS method applied to $A \in \mathbb{R}^{m\times n}$ $(m \ge n)$ compute a QR factorization $A = QR$, $Q \in \mathbb{R}^{m\times n}$. Define the orthogonal projection $P_i = q_i q_i^T$, where $q_i = Q(:, i)$. Show that

$$(I - P_j)(I - P_{j-1}) \dots (I - P_1) = I - P_j - \cdots - P_1.$$

Show that the CGS method corresponds to the operations

$$a_j \leftarrow (I - P_j - \cdots - P_1)a_j, \qquad j = 1\colon n,$$

while MGS corresponds to

$$a_j \leftarrow (I - P_j)(I - P_{j-1}) \dots (I - P_1)a_j, \qquad j = 1\colon n.$$

19.9. Let $A = [a_1, a_2] \in \mathbb{R}^{m\times 2}$ and denote the angle between a_1 and a_2 by θ, $0 \le \theta \le \pi/2$. (Thus, $\cos\theta := |a_1^T a_2|/(\|a_1\|_2 \|a_2\|_2)$.) Show that

$$\kappa_2(A) \ge \frac{\max(\|a_1\|_2, \|a_2\|_2)}{\min(\|a_1\|_2, \|a_2\|_2)} \cot\theta.$$

19.10. (Björck [119, 1967]) Let

$$A = \begin{bmatrix} 1 & 1 & 1 \\ \epsilon & 0 & 0 \\ 0 & \epsilon & 0 \\ 0 & 0 & \epsilon \end{bmatrix},$$

which is a matrix of the form discussed by Läuchli [772, 1961]. Assuming that $fl(1 + \epsilon^2) = 1$, evaluate the Q matrices produced by the CGS and MGS methods and assess their orthonormality.

19.11. Show that the matrix P in (19.27) has the form

$$P = \begin{bmatrix} 0_n & Q^T \\ Q & I - QQ^T \end{bmatrix},$$

where Q is the matrix obtained from the MGS method applied to A.

19.12. (Björck and Paige [131, 1992]) For any matrices satisfying

$$\begin{bmatrix} \Delta A_1 \\ A + \Delta A_2 \end{bmatrix} = \begin{bmatrix} P_{11} \\ P_{21} \end{bmatrix} R, \qquad P_{11}^T P_{11} + P_{21}^T P_{21} = I,$$

where both P_{11} and P_{21} have at least as many rows as columns, show that there exists an orthonormal Q such that $A + \Delta A = QR$, where

$$\Delta A = F \Delta A_1 + \Delta A_2, \qquad \|F\|_2 \leq 1.$$

(Hint: use the CS decomposition $P_{11} = UCW^T$, $P_{21} = VSW^T$, where U and V have orthonormal columns, W is orthogonal, and C and S are square, nonnegative diagonal matrices with $C^2 + S^2 = I$. Let $Q = VW^T$. Note, incidentally, that $P_{21} = VW^T \cdot WSW^T$, so $Q = VW^T$ is the orthonormal polar factor of P_{21} and hence is the nearest orthonormal matrix to P_{21} in the 2- and Frobenius norms. For details of the CS decomposition see Golub and Van Loan [509, 1996, §§2.6.4, 8.7.3] and Paige and Wei [913, 1994].)

19.13. We know that Householder QR factorization of $\left[\begin{smallmatrix} 0 \\ A \end{smallmatrix}\right]$ is equivalent to the MGS method applied to A, and Problem 19.11 shows that the orthonormal matrix Q from MGS is a submatrix of the orthogonal matrix P from the Householder method. Since Householder's method produces a nearly orthogonal P, does it not follow that MGS must also produce a nearly orthonormal Q?

19.14. (Higham [603, 1994]) Let $A \in \mathbb{R}^{m \times n}$ $(m \geq n)$ have the polar decomposition $A = UH$. Show that

$$\frac{\|A^TA - I\|_2}{1 + \|A\|_2} \leq \|A - U\|_2 \leq \frac{\|A^TA - I\|_2}{1 + \sigma_{\min}(A)}.$$

This result shows that the two measures of orthonormality $\|A^TA - I\|_2$ and $\|A - U\|_2$ are essentially equivalent (cf. (19.30)).