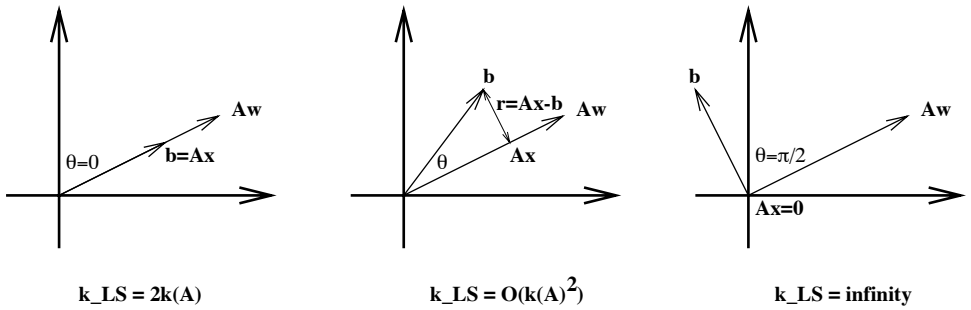


An alternative form for the bound in Theorem 3.4 that eliminates the $O(\epsilon^2)$ term is as follows [256, 147] (here $\tilde{r}$ is the perturbed residual $\tilde{r} = (b + \delta b) - (A + \delta A)\tilde{x}$):

$$\begin{aligned} \frac{\|\tilde{x} - x\|_2}{\|x\|_2} &\leq \frac{\epsilon\kappa_2(A)}{1 - \epsilon\kappa_2(A)} \left(2 + (\kappa_2(A) + 1)\frac{\|r\|_2}{\|A\|_2\|x\|_2} \right), \\ \frac{\|\tilde{r} - r\|_2}{\|r\|_2} &\leq (1 + 2\epsilon\kappa_2(A)). \end{aligned}$$

We will see that, properly implemented, both the QR decomposition and SVD are numerically stable; i.e., they yield a solution $\tilde{x}$ minimizing $\|(A + \delta A)\tilde{x} - (b + \delta b)\|_2$ with

$$\max\left(\frac{\|\delta A\|}{\|A\|}, \frac{\|\delta b\|}{\|b\|} \right) = O(\varepsilon).$$

We may combine this with the above perturbation bounds to get error bounds for the solution of the least squares problem, much as we did for linear equation solving.

The normal equations are not as accurate. Since they involve solving $(A^T A)x = A^T b$, the accuracy depends on the condition number $\kappa_2(A^T A) = \kappa_2^2(A)$. Thus the error is always bounded by on $\kappa_2^2(A)\varepsilon$, never just $\kappa_2(A)\varepsilon$. Therefore we expect that the normal equations can lose twice as many digits of accuracy as methods based on the QR decomposition and SVD.

Furthermore, solving the normal equations is *not* necessarily stable; i.e., the computed solution $\tilde{x}$ does not generally minimize $\|(A + \delta A)\tilde{x} - (b + \delta b)\|_2$ for small δA and δb. Still, when the condition number is small, we expect the normal equations to be about as accurate as the QR decomposition or SVD. Since the normal equations are the fastest way to solve the least squares problem, they are the method of choice when the matrix is well-conditioned.

We return to the problem of solving very ill-conditioned least squares problems in section 3.5.

3.4. Orthogonal Matrices

As we said in section 3.2.2, Gram–Schmidt orthogonalization (Algorithm 3.1) may not compute an orthogonal matrix Q when the vectors being orthogonal-

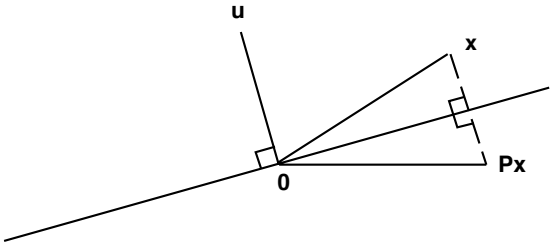
ized are nearly linearly dependent, so we cannot use it to compute the QR decomposition stably.

Instead, we base our algorithms on certain easily computable orthogonal matrices called *Householder reflections* and *Givens rotations*, which we can choose to introduce zeros into vectors that they multiply. Later we will show that any algorithm that uses these orthogonal matrices to introduce zeros is automatically stable. This error analysis will apply to our algorithms for the QR decomposition as well as many SVD and eigenvalue algorithms in Chapters 4 and 5.

Despite the possibility of nonorthogonal Q, the MGS algorithm has important uses in numerical linear algebra. (There is little use for its less stable version, CGS.) These uses include finding eigenvectors of symmetric tridiagonal matrices using bisection and inverse iteration (section 5.3.4) and the Arnoldi and Lanczos algorithms for reducing a matrix to certain "condensed" forms (sections 6.6.1, 6.6.6, and 7.4). Arnoldi and Lanczos are used as the basis of algorithms for solving sparse linear systems and finding eigenvalues of sparse matrices. MGS can also be modified to solve the least squares problem stably, but Q may still be far from orthogonal [33].

3.4.1. Householder Transformations

A Householder transformation (or reflection) is a matrix of the form $P = I - 2uu^T$ where $\|u\|_2 = 1$. It is easy to see that $P = P^T$ and $PP^T = (I - 2uu^T)(I - 2uu^T) = I - 4uu^T + 4uu^Tuu^T = I$, so P is a symmetric, orthogonal matrix. It is called a reflection because Px is reflection of x in the plane through 0 perpendicular to u.



Given a vector x, it is easy to find a Householder reflection $P = I - 2uu^T$ to zero out all but the first entry of x: $Px = [c, 0, \ldots, 0]^T = c \cdot e_1$. We do this as follows. Write $Px = x - 2u(u^Tx) = c \cdot e_1$ so that $u = \frac{1}{2(u^Tx)}(x - ce_1)$; i.e., u is a linear combination of x and e_1. Since $\|x\|_2 = \|Px\|_2 = |c|$, u must be parallel to the vector $\tilde{u} = x \pm \|x\|_2 e_1$, and so $u = \tilde{u}/\|\tilde{u}\|_2$. One can verify that either choice of sign yields a u satisfying $Px = ce_1$, as long as $\tilde{u} = 0$. We will use $\tilde{u} = x + \text{sign}(x_1)e_1$, since this means that there is no cancellation in

computing the first component of $\tilde{u}$. In summary, we get

$$\tilde{u} = \begin{bmatrix} x_1 + \text{sign}(x_1) \cdot \|x\|_2 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{with} \quad u = \frac{\tilde{u}}{\|\tilde{u}\|_2}.$$

We write this as $u = \text{House}(x)$. (In practice, we can store $\tilde{u}$ instead of u to save the work of computing u, and use the formula $P = I - (2/\|\tilde{u}\|_2^2)\tilde{u}\tilde{u}^T$ instead of $P = I - 2uu^T$.)

EXAMPLE 3.5. We show how to compute the QR decomposition of a a 5-by-4 matrix A using Householder transformations. This example will make the pattern for general m-by-n matrices evident. In the matrices below, P_i is a 5-by-5 orthogonal matrix, x denotes a generic nonzero entry, and o denotes a zero entry.

1. Choose P_1 so $$A_1 \equiv P_1 A = \begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & x & x & x \\ o & x & x & x \\ o & x & x & x \end{bmatrix}.$$

2. Choose $P_2 = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & P_2' \end{array}\right]$ so $$A_2 \equiv P_2 A_1 = \begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & o & x & x \\ o & o & x & x \\ o & o & x & x \end{bmatrix}.$$

3. Choose $P_3 = \left[\begin{array}{cc|c} 1 & & \\ & 1 & 0 \\ \hline & 0 & P_3' \end{array}\right]$ so $$A_3 \equiv P_3 A_2 = \begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & o & x & x \\ o & o & o & x \\ o & o & o & x \end{bmatrix}.$$

4. Choose $P_4 = \left[\begin{array}{ccc|c} 1 & & & \\ & 1 & & 0 \\ & & 1 & \\ \hline & 0 & & P_4' \end{array}\right]$ so $$A_4 \equiv P_4 A_3 = \begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & o & x & x \\ o & o & o & x \\ o & o & o & o \end{bmatrix}.$$

Here, we have chosen a Householder matrix P_i' to zero out the subdiagonal entries in column i; this does not disturb the zeros already introduced in previous columns.

Let us call the final 5-by-4 upper triangular matrix $\tilde{R} \equiv A_4$. Then $A = P_1^T P_2^T P_3^T P_4^T \tilde{R} = QR$, where Q is the first four columns of $P_1^T P_2^T P_3^T P_4^T = P_1 P_2 P_3 P_4$ (since all P_i are symmetric) and R is the first four rows of $\tilde{R}$. $\diamond$

Here is the general algorithm for QR decomposition using Householder transformations.

ALGORITHM 3.2. *QR factorization using Householder reflections:*

for $i = 1$ *to* n
$u_i = \text{House}(A(i:m, i))$
$P_i = I - 2u_i u_i^T$
$A(i:m, i:n) = P_i A(i:m, i:n)$
end for

Here are some more implementation details. We never need to form P_i explicitly but just multiply

$$(I - 2u_i u_i^T)A(i:m, i:n) = A(i:m, i:n) - 2u_i(u_i^T A(i:m, i:n)),$$

which costs less. To store P_i, we need only u_i, or $\tilde{u}_i$ and $\|\tilde{u}_i\|$. These can be stored in column i of A; in fact it need not be changed! Thus QR can be "overwritten" on A, where Q is stored in factored form $P_1 \cdots P_{n-1}$, and P_i is stored as $\tilde{u}_i$ below the diagonal in column i of A. (We need an extra array of length n for the top entry of $\tilde{u}_i$, since the diagonal entry is occupied by R_{ii}.)

Recall that to solve the least squares problem $\min \|Ax - b\|_2$ using $A = QR$, we need to compute $Q^T b$. This is done as follows: $Q^T b = P_n P_{n-1} \cdots P_1 b$, so we need only keep multiplying b by $P_1, P_2, \ldots, P_n$:

for $i = 1$ *to* n
$\gamma = -2 \cdot u_i^T b$
$b = b + \gamma u_i$
end for

The cost is n dot products $\gamma = -2 \cdot u_i^T b$ and n "saxpys" $b + \gamma u_i$. The cost of computing $A = QR$ this way is $2n^2 m - \frac{2}{3}n^3$, and the subsequent cost of solving the least squares problem given QR is just an additional $O(mn)$.

The LAPACK routine for solving the least squares problem using QR is `sgels`. Just as Gaussian elimination can be reorganized to use matrix-matrix multiplication and other Level 3 BLAS (see section 2.6), the same can be done for the QR decomposition; see Question 3.17. In Matlab, if the m-by-n matrix A has more rows than columns and b is m by 1, `A\b` solves the least squares problem. The QR decomposition itself is also available via [Q,R]=qr(A).

3.4.2. Givens Rotations

A Givens rotation $R(\theta) \equiv \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$ rotates any vector $x \in \mathbb{R}^2$ counterclockwise by θ: