Chapter 4

Column Spaces and QR

One way to interpret the linear problem $A\vec{x} = \vec{b}$ for $\vec{x}$ is that we wish to write $\vec{b}$ as a linear combination of the columns of A with weights given in $\vec{x}$. This perspective does not change when we allow $A \in \mathbb{R}^{m\times n}$ to be non-square, but the solution may not exist or be unique depending on the structure of the column space. For these reasons, some techniques for factoring matrices and analyzing linear systems seek simpler *representations* of the column space to disambiguate solvability and span more explicitly than row-based factorizations like LU.

4.1 The Structure of the Normal Equations

As we have shown, a necessary and sufficient condition for $\vec{x}$ to be a solution of the least-squares problem $A\vec{x} \approx \vec{b}$ is that $\vec{x}$ satisfy the normal equations $(A^\top A)\vec{x} = A^\top \vec{b}$. This theorem suggests that solving least-squares is a simple enough extension of linear techniques. Methods such as Cholesky factorization also show that the special structure of least-squares problems can be used to the solver's advantage.

There is one large problem limiting the use of this approach, however. For now, suppose A is square; then we can write:

$$\begin{aligned}
\text{cond } A^\top A &= \|A^\top A\| \|(A^\top A)^{-1}\| \\
&\approx \|A^\top\| \|A\| \|A^{-1}\| \|(A^\top)^{-1}\| \text{ depending on the choice of } \|\cdot\| \\
&= \|A\|^2 \|A^{-1}\|^2 \\
&= (\text{cond } A)^2
\end{aligned}$$

That is, the condition number of $A^\top A$ is approximately the **square** of the condition number of A! Thus, while generic linear strategies might work on $A^\top A$ when the least-squares problem is "easy," when the columns of A are nearly linearly dependent these strategies are likely to generate considerable error since they do not deal with A directly.

Intuitively, a primary reason that cond A can be large is that columns of A might look "similar." Think of each column of A as a vector in $\mathbb{R}^m$. If two columns $\vec{a}_i$ and $\vec{a}_j$ satisfy $\vec{a}_i \approx \vec{a}_j$, then the least-squares residual length $\|\vec{b} - A\vec{x}\|$ probably would not suffer much if we replace multiples of $\vec{a}_i$ with multiples of $\vec{a}_j$ or vice versa. This wide range of nearly–but not completely–equivalent solutions yields poor conditioning. While the resulting vector $\vec{x}$ is unstable, however, the product

$A\vec{x}$ remains nearly unchanged, by design of our substitution. Therefore, if we wish to solve $A\vec{x} \approx \vec{b}$ simply to write $\vec{b}$ in the column space of A, either solution would suffice.

To solve such poorly-conditioned problems, we will employ an alternative strategy with closer attention to the column space of A rather than employing row operations as in Gaussian elimination. This way, we can identify such near-dependencies *explicitly* and deal with them in a numerically stable way.

4.2 Orthogonality

We have determined when the least-squares problem is difficult, but we might also ask when it is most straightforward. If we can reduce a system to the straightforward case without inducing conditioning problems along the way, we will have found a more stable way around the issues explained in §4.1.

Obviously the easiest linear system to solve is $I_{n\times n}\vec{x} = \vec{b}$: The solution simply is $\vec{x} \equiv \vec{b}$! We are unlikely to enter this particular linear system into our solver explicitly, but we may do so accidentally while solving least-squares. In particular, even when $A \neq I_{n\times n}$—in fact, A need not be a square matrix—we may in particularly lucky circumstances find that the normal matrix $A^\top A$ satisfies $A^\top A = I_{n\times n}$. To avoid confusion with the general case, we will use the letter Q to represent such a matrix.

Simply praying that $Q^\top Q = I_{n\times n}$ unlikely will yield a desirable solution strategy, but we can examine this case to see how it becomes so favorable. Write the columns of Q as vectors $\vec{q}_1, \cdots, \vec{q}_n \in \mathbb{R}^m$. Then, it is easy to verify that the product $Q^\top Q$ has the following structure:

$$Q^\top Q = \begin{pmatrix} - & \vec{q}_1^\top & - \\ - & \vec{q}_2^\top & - \\ & \vdots & \\ - & \vec{q}_n^\top & \end{pmatrix} \begin{pmatrix} | & | & & | \\ \vec{q}_1 & \vec{q}_2 & \cdots & \vec{q}_n \\ | & | & & | \end{pmatrix} = \begin{pmatrix} \vec{q}_1\cdot\vec{q}_1 & \vec{q}_1\cdot\vec{q}_2 & \cdots & \vec{q}_1\cdot\vec{q}_n \\ \vec{q}_2\cdot\vec{q}_1 & \vec{q}_2\cdot\vec{q}_2 & \cdots & \vec{q}_2\cdot\vec{q}_n \\ \vdots & \vdots & \cdots & \vdots \\ \vec{q}_n\cdot\vec{q}_1 & \vec{q}_n\cdot\vec{q}_2 & \cdots & \vec{q}_n\cdot\vec{q}_n \end{pmatrix}$$

Setting the expression on the right equal to $I_{n\times n}$ yields the following relationship:

$$\vec{q}_i \cdot \vec{q}_j = \begin{cases} 1 & \text{when } i = j \\ 0 & \text{when } i \neq j \end{cases}$$

In other words, the columns of Q are unit-length and orthogonal to one another. We say that they form an *orthonormal basis* for the column space of Q:

Definition 4.1 (Orthonormal; orthogonal matrix)**.** *A set of vectors $\{\vec{v}_1, \cdots, \vec{v}_k\}$ is* orthonormal *if $\|\vec{v}_i\| = 1$ for all i and $\vec{v}_i \cdot \vec{v}_j = 0$ for all $i \neq j$. A square matrix whose columns are orthonormal is called an* orthogonal *matrix.*

We motivated our discussion by asking when we can expect $Q^\top Q = I_{n\times n}$. Now it is easy to see that this occurs when the columns of Q are orthonormal. Furthermore, if Q is square and invertible with $Q^\top Q = I_{n\times n}$, then simply by multiplying both sides of this expression by Q^{-1} we find $Q^{-1} = Q^\top$. Thus, solving $Q\vec{x} = \vec{b}$ in this case is as easy as multiplying both sides by the transpose $Q^\top$.

Orthonormality also has a strong geometric interpretation. Recall from Chapter 0 that we can regard two orthogonal vectors $\vec{a}$ and $\vec{b}$ as being *perpendicular*. So, an orthonormal set of vectors simply is a set of unit-length perpendicular vectors in $\mathbb{R}^n$. If Q is orthogonal, then its action does not affect the length of vectors:

$$\|Q\vec{x}\|^2 = \vec{x}^\top Q^\top Q\vec{x} = \vec{x}^\top I_{n\times n}\vec{x} = \vec{x}\cdot\vec{x} = \|\vec{x}\|^2$$

Similarly, Q cannot affect the angle between two vectors, since:

$$(Q\vec{x})\cdot(Q\vec{y}) = \vec{x}^\top Q^\top Q\vec{y} = \vec{x}^\top I_{n\times n}\vec{y} = \vec{x}\cdot\vec{y}$$

From this standpoint, if Q is orthogonal, then Q represents an *isometry* of $\mathbb{R}^n$, that is, it preserves lengths and angles. In other words, it can rotate or reflect vectors but cannot scale or shear them. From a high level, the linear algebra of orthogonal matrices is easier because their action does not affect the geometry of the underlying space in any nontrivial way.

4.2.1 Strategy for Non-Orthogonal Matrices

Except in special circumstances, most of our matrices A when solving $A\vec{x} = \vec{b}$ or the corresponding least-squares problem will not be orthogonal, so the machinery of §4.2 does not apply directly. For this reason, we must do some additional computations to connect the general case to the orthogonal one.

Take a matrix $A \in \mathbb{R}^{m\times n}$, and denote its column space as col A; recall that col A represents the span of the columns of A. Now, suppose a matrix $B \in \mathbb{R}^{n\times n}$ is invertible. We can make a simple observation about the column space of AB relative to that of A:

Lemma 4.1 (Column space invariance)**.** *For any $A \in \mathbb{R}^{m\times n}$ and invertible $B \in \mathbb{R}^{n\times n}$,*

$$\textit{col } A = \textit{col } AB.$$

Proof. Suppose $\vec{b} \in$ col A. Then, by definition of multiplication by A there exists $\vec{x}$ with $A\vec{x} = \vec{b}$. Then, $(AB)\cdot(B^{-1}\vec{x}) = A\vec{x} = \vec{b}$, so $\vec{b} \in$ col AB. Conversely, take $\vec{c} \in$ col AB, so there exists $\vec{y}$ with $(AB)\vec{y} = \vec{c}$. Then, $A\cdot(B\vec{y}) = \vec{c}$, showing that $\vec{c}$ is in col A. □

Recall the "elimination matrix" description of Gaussian elimination: We started with a matrix A and applied row operation matrices E_i such that the sequence $A, E_1A, E_2E_1A, \ldots$ represented sequentially easier linear systems. The lemma above suggests an alternative strategy for situations in which we care about the column space: Apply *column* operations to A by *post*-multiplication until the columns are orthonormal. That is, we obtain a product $Q = AE_1E_2\cdots E_k$ such that Q is orthonormal. As long as the E_i's are invertible, the lemma shows that col Q = col A. Inverting these operations yields a factorization $A = QR$ for $R = E_k^{-1}E_{k-1}^{-1}\cdots E_1^{-1}$.

As in the LU factorization, if we design R carefully, the solution of least-squares problems $A\vec{x} \approx \vec{b}$ may simplify. In particular, when $A = QR$, we can write the solution to $A^\top A\vec{x} = A^\top\vec{b}$ as follows:

$$\begin{aligned}
\vec{x} &= (A^\top A)^{-1}A^\top\vec{b} \\
&= (R^\top Q^\top QR)^{-1}R^\top Q^\top\vec{b} \text{ since } A = QR \\
&= (R^\top R)^{-1}R^\top Q^\top\vec{b} \text{ since } Q \text{ is orthogonal} \\
&= R^{-1}(R^\top)^{-1}R^\top Q^\top\vec{b} \text{ since } (AB)^{-1} = B^{-1}A^{-1} \\
&= R^{-1}Q^\top\vec{b}
\end{aligned}$$

Or equivalently, $R\vec{x} = Q^\top\vec{b}$

Thus, if we design R to be a triangular matrix, then solving the linear system $R\vec{x} = Q^\top \vec{b}$ is as simple as back-substitution.

Our task for the remainder of the chapter is to design strategies for such a factorization.

4.3 Gram-Schmidt Orthogonalization

Our first approach for finding QR factorizations is the simplest to describe and implement but may suffer from numerical issues. We use it here as an initial strategy and then will improve upon it with better operations.

4.3.1 Projections

Suppose we have two vectors $\vec{a}$ and $\vec{b}$. Then, we could easily ask "Which multiple of $\vec{a}$ is closest to $\vec{b}$?" Mathematically, this task is equivalent to minimizing $\|c\vec{a} - \vec{b}\|^2$ over all possible $c \in \mathbb{R}$. If we think of $\vec{a}$ and $\vec{b}$ as $n \times 1$ matrices and c as a 1×1 matrix, then this is nothing more than an unconventional least-squares problem $\vec{a} \cdot c \approx \vec{b}$. In this case, the normal equations show $\vec{a}^\top \vec{a} \cdot c = \vec{a}^\top \vec{b}$, or

$$c = \frac{\vec{a} \cdot \vec{b}}{\vec{a} \cdot \vec{a}} = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|^2}.$$

We denote this *projection* of $\vec{b}$ onto $\vec{a}$ as:

$$\mathrm{proj}_{\vec{a}}\, \vec{b} = c\vec{a} = \frac{\vec{a} \cdot \vec{b}}{\vec{a} \cdot \vec{a}} \vec{a} = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|^2} \vec{a}$$

Obviously $\mathrm{proj}_{\vec{a}}\, \vec{b}$ is parallel to $\vec{a}$. What about the remainder $\vec{b} - \mathrm{proj}_{\vec{a}} \vec{b}$? We can do a simple computation to find out:

$$\begin{aligned}
\vec{a} \cdot (\vec{b} - \mathrm{proj}_{\vec{a}}\, \vec{b}) &= \vec{a} \cdot \vec{b} - \vec{a} \cdot \left(\frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|^2} \vec{a} \right) \\
&= \vec{a} \cdot \vec{b} - \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|^2} (\vec{a} \cdot \vec{a}) \\
&= \vec{a} \cdot \vec{b} - \vec{a} \cdot \vec{b} \\
&= 0
\end{aligned}$$

Thus, we have decomposed $\vec{b}$ into a component parallel to $\vec{a}$ and another component orthogonal to $\vec{a}$.

Now, suppose that $\hat{a}_1, \hat{a}_2, \cdots, \hat{a}_k$ are orthonormal; we will put hats over vectors with unit length. Then, for any single i we can see:

$$\mathrm{proj}_{\hat{a}_i}\, \vec{b} = (\hat{a}_i \cdot \vec{b}) \hat{a}_i$$

The norm term does not appear because $\|\hat{a}_i\| = 1$ by definition. We could project $\vec{b}$ onto span $\{\hat{a}_1, \cdots, \hat{a}_k\}$ by minimizing the following energy over $c_1, \ldots, c_k \in \mathbb{R}$:

$$\begin{aligned}\|c_1\hat{a}_1 + c_2\hat{a}_2 + \cdots + c_k\hat{a}_k - \vec{b}\|^2 &= \left(\sum_{i=1}^{k}\sum_{j=1}^{k} c_i c_j (\hat{a}_i \cdot \hat{a}_j)\right) - 2\vec{b} \cdot \left(\sum_{i=1}^{k} c_i \hat{a}_i\right) + \vec{b} \cdot \vec{b} \\ &\quad \text{by applying } \|\vec{v}\|^2 = \vec{v} \cdot \vec{v} \\ &= \sum_{i=1}^{k}\left(c_i^2 - 2c_i\vec{b} \cdot \hat{a}_i\right) + \|\vec{b}\|^2 \text{ since the } \hat{a}_i\text{'s are orthonormal}\end{aligned}$$

Notice that the second step here is only valid because of orthonormality. At a minimum, the derivative with respect to c_i is zero for every c_i, yielding:

$$c_i = \hat{a}_i \cdot \vec{b}$$

Thus, we have shown that when $\hat{a}_1, \cdots, \hat{a}_k$ are orthonormal, the following relationship holds:

$$\text{proj}_{\text{span }\{\hat{a}_1, \cdots, \hat{a}_k\}} \vec{b} = (\hat{a}_1 \cdot \vec{b})\hat{a}_1 + \cdots + (\hat{a}_k \cdot \vec{b})\hat{a}_k$$

This is simply an extension of our projection formula, and by a similar proof it is easy to see that

$$\hat{a}_i \cdot (\vec{b} - \text{proj}_{\text{span }\{\hat{a}_1, \cdots, \hat{a}_k\}} \vec{b}) = 0.$$

That is, we have separated $\vec{b}$ into a component parallel to the span of the $\hat{a}_i$'s and a perpendicular residual.

4.3.2 Gram-Schmidt Orthogonalization

Our observations above lead to a simple algorithm for *orthogonalization*, or finding an orthogonal basis $\{\hat{a}_1, \cdots, \hat{a}_k\}$ whose span is the same as that of a set of linearly independent input vectors $\{\vec{v}_1, \cdots, \vec{v}_k\}$:

1. Set
$$\hat{a}_1 \equiv \frac{\vec{v}_1}{\|\vec{v}_1\|}.$$
   That is, we take $\hat{a}_1$ to be a unit vector parallel to $\vec{v}_1$.

2. For i from 2 to k,

   (a) Compute the projection
   $$\vec{p}_i \equiv \text{proj}_{\text{span }\{\hat{a}_1, \cdots, \hat{a}_{i-1}\}} \vec{v}_i.$$
   By definition $\hat{a}_1, \cdots, \hat{a}_{i-1}$ are orthonormal, so our formula above applies.

   (b) Define
   $$\hat{a}_i \equiv \frac{\vec{v}_i - \vec{p}_i}{\|\vec{v}_i - \vec{p}_i\|}.$$

This technique, known as "Gram-Schmidt Orthogonalization," is a straightforward application of our discussion above. The key to the proof of this technique is to notice that span $\{\vec{v}_1, \cdots, \vec{v}_i\} =$ span $\{\hat{a}_1, \cdots, \hat{a}_i\}$ for each $i \in \{1, \cdots, k\}$. Step 1 clearly makes this the case for $i = 1$, and for $i > 1$ the definition of $\hat{a}_i$ in step 2b simply removes the projection onto the vectors we already have seen.

If we start with a matrix A whose columns are $\vec{v}_1, \cdots, \vec{v}_k$, then we can implement Gram-Schmidt as a series of column operations on A. Dividing column i of A by its norm is equivalent to post-multiplying A by a $k \times k$ diagonal matrix. Similarly, subtracting off the projection of a column onto the orthonormal columns to its left as in step 2 is equivalent to post-multiplying by an upper-triangular matrix: Be sure to understand why this is the case! Thus, our discussion in §4.2.1 applies, and we can use Gram-Schmidt to obtain a factorization $A = QR$.

Unfortunately, the Gram-Schmidt algorithm can introduce serious numerical instabilities due to the subtraction step. For instance, suppose we provide the vectors $\vec{v}_1 = (1, 1)$ and $\vec{v}_2 = (1 + \varepsilon, 1)$ as input to Gram-Schmidt for some $0 < \varepsilon \ll 1$. Notice that an obvious basis for span $\{\vec{v}_1, \vec{v}_2\}$ is $\{(1, 0), (0, 1)\}$. But, if we apply Gram-Schmidt, we obtain:

$$\begin{aligned}
\hat{a}_1 &= \frac{\vec{v}_1}{\|\vec{v}_1\|} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\
\vec{p}_2 &= \frac{2+\varepsilon}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\
\vec{v}_2 - \vec{p}_2 &= \begin{pmatrix} 1+\varepsilon \\ 1 \end{pmatrix} - \frac{2+\varepsilon}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\
&= \frac{1}{2} \begin{pmatrix} \varepsilon \\ -\varepsilon \end{pmatrix}
\end{aligned}$$

Notice that $\|\vec{v}_2 - \vec{p}_2\| = (\sqrt{2}/2) \cdot \varepsilon$, so computing $\hat{a}_2$ will require division by a scalar on the order of ε. Division by small numbers is an unstable numerical operation that we should avoid.

4.4 Householder Transformations

In §4.2.1, we motivated the construction of QR factorization by post-multiplication and column operations. This construction is reasonable in the context of analyzing column spaces, but as we saw in our derivation of the Gram-Schmidt algorithm, the resulting numerical techniques can be unstable.

Rather than starting with A and post-multiplying by column operations to obtain $Q = AE_1 \cdots E_k$, however, we can preserve our high-level strategy from Gaussian elimination. That is, we can start with A and *pre*-multiply by orthogonal matrices Q_i to obtain $Q_k \cdots Q_1 A = R$; these Q's will act like row operations, eliminating elements of A until the resulting product R is upper-triangular. Then, thanks to orthogonality of the Q's we can write $A = Q_1^\top \cdots Q_k^\top R$, obtaining the QR factorization since the product of orthogonal matrices is orthogonal.

The row operation matrices we used in Gaussian elimination and LU will not suffice for QR factorization since they are not orthogonal. A number of alternatives have been suggested; we will introduce one common strategy introduced in 1958 by Alston Scott Householder.

The space of orthogonal $n \times n$ matrices is very large, so we must find a smaller space of Q_i's that is easier to work with. From our geometric discussions in §4.2, we know that orthogonal matrices must preserve angles and lengths, so intuitively they only can rotate and reflect vectors.

Thankfully, the reflections can be easy to write in terms of projections, as illustrated in Figure NUMBER. Suppose we have a vector $\vec{b}$ that we wish to reflect over a vector $\vec{v}$. We have shown that the residual $\vec{r} \equiv \vec{b} - \text{proj}_{\vec{v}}\vec{b}$ is perpendicular to $\vec{v}$. As in Figure NUMBER, the difference $2\text{proj}_{\vec{v}}\vec{b} - \vec{b}$ reflects $\vec{b}$ *over* $\vec{v}$.

We can expand our reflection formula as follows:

$$\begin{aligned}
2\text{proj}_{\vec{v}}\vec{b} - \vec{b} &= 2\frac{\vec{v}\cdot\vec{b}}{\vec{v}\cdot\vec{v}}\vec{v} - \vec{b} \text{ by definition of projection} \\
&= 2\vec{v}\cdot\frac{\vec{v}^\top\vec{b}}{\vec{v}^\top\vec{v}} - \vec{b} \text{ using matrix notation} \\
&= \left(\frac{2\vec{v}\vec{v}^\top}{\vec{v}^\top v} - I_{n\times n}\right)\vec{b} \\
&\equiv -H_{\vec{v}}\vec{b} \text{ where the negative is introduced to align with other treatments}
\end{aligned}$$

Thus, we can think of reflecting $\vec{b}$ over $\vec{v}$ as applying a linear operator $-H_{\vec{v}}$ to $\vec{b}$! Of course, $H_{\vec{v}}$ without the negative is still orthogonal, so we will use it from now on.

Suppose we are doing the first step of forward substitution during Gaussian elimination. Then, we wish to pre-multiply A by a matrix that takes the first column of A, which we will denote $\vec{a}$, to some multiple of the first identity vector $\vec{e}_1$. In other words, we want for some $c \in \mathbb{R}$:

$$\begin{aligned}
c\vec{e}_1 &= H_{\vec{v}}\vec{a} \\
&= \left(I_{n\times n} - \frac{2\vec{v}\vec{v}^\top}{\vec{v}^\top v}\right)\vec{a} \\
&= \vec{a} - 2\vec{v}\frac{\vec{v}^\top\vec{a}}{\vec{v}^\top v}
\end{aligned}$$

Moving terms around shows

$$\vec{v} = (\vec{a} - c\vec{e}_1)\cdot\frac{\vec{v}^\top\vec{v}}{2\vec{v}^\top\vec{a}}$$

In other words, $\vec{v}$ must be parallel to the difference $\vec{a} - c\vec{e}_1$. In fact, scaling $\vec{v}$ does not affect the formula for $H_{\vec{v}}$, so we can *choose* $\vec{v} = \vec{a} - c\vec{e}_1$. Then, for our relationship to hold we must have

$$\begin{aligned}
1 &= \frac{\vec{v}^\top\vec{v}}{2\vec{v}^\top\vec{a}} \\
&= \frac{\|\vec{a}\|^2 - 2c\vec{e}_1\cdot a + c^2}{2(\vec{a}\cdot\vec{a} - c\vec{e}_1\cdot\vec{a})} \\
\text{Or, } 0 &= \|\vec{a}\|^2 - c^2 \implies c = \pm\|\vec{a}\|
\end{aligned}$$

With this choice of c, we have shown:

$$H_{\vec{v}}A = \begin{pmatrix} c & \times & \times & \times \\ 0 & \times & \times & \times \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \times & \times & \times \end{pmatrix}$$

We have just accomplished a step akin to forward elimination using only orthogonal matrices!

Proceeding, in the notation of CITE during the k-th step of triangularization we have a vector $\vec{a}$ that we can split into two components:

$$\vec{a} = \begin{pmatrix} \vec{a}_1 \\ \vec{a}_2 \end{pmatrix}$$

Here, $\vec{a}_1 \in \mathbb{R}^k$ and $\vec{a}_2 \in \mathbb{R}^{m-k}$. We wish to find $\vec{v}$ such that

$$H_{\vec{v}}\vec{a} = \begin{pmatrix} \vec{a}_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Following a parallel derivation to the one above, it is easy to show that

$$\vec{v} = \begin{pmatrix} \vec{0} \\ \vec{a}_2 \end{pmatrix} - c\vec{e}_k$$

accomplishes exactly this transformation when $c = \pm\|\vec{a}_2\|$; we usually choose the sign of c to avoid cancellation by making it have sign opposite to that of the k-th value in $\vec{a}$.

The algorithm for Householder QR is thus fairly straightforward. For each column of A, we compute $\vec{v}$ annihilating the bottom elements of the column and apply $H_{\vec{v}}$ to A. The end result is an upper triangular matrix $R = H_{\vec{v}_n} \cdots H_{\vec{v}_1} A$. The orthogonal matrix Q is given by the product $H_{\vec{v}_1}^\top \cdots H_{\vec{v}_n}^\top$, which can be stored implicitly as a list of vectors $\vec{v}$, which fits in the lower triangle as shown above.

4.5 Reduced QR Factorization

We conclude our discussion by returning to the most general case $A\vec{x} \approx \vec{b}$ when $A \in \mathbb{R}^{m\times n}$ is not square. Notice that both algorithms we have discussed in this chapter can factor non-square matrices A into products QR, but the output is somewhat different:

- When applying Gram-Schmidt, we do column operations on A to obtain Q by orthogonalization. For this reason, the dimension of A is that of Q, yielding $Q \in \mathbb{R}^{m\times n}$ and $R \in \mathbb{R}^{n\times n}$.
- When using Householder reflections, we obtain Q as the product of a number of $m \times m$ reflection matrices, leaving $R \in \mathbb{R}^{m\times n}$.

Suppose we are in the typical case for least-squares, for which $m \gg n$. We still prefer to use the Householder method due to its numerical stability, but now the $m \times m$ matrix Q might be too large to store! Thankfully, we know that R is upper triangular. For instance, consider the structure of a 5×3 matrix R:

$$R = \left(\begin{array}{ccc} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right)$$

It is easy to see that anything below the upper $n \times n$ square of R must be zero, yielding a simplification:

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

Here, $Q_1 \in \mathbb{R}^{m \times n}$ and $R_1 \in \mathbb{R}^{n \times n}$ still contains the upper triangle of R. This is called the "reduced" QR factorization of A, since the columns of Q_1 contain a basis for the column space of A rather than for all of $\mathbb{R}^m$; it takes up far less space. Notice that the discussion in §4.2.1 still applies, so the reduced QR factorization can be used for least-squares in a similar fashion.

4.6 Problems

- tridiagonalization with Householder
- Givens
- Underdetermined QR