3 Gram-Schmidt Orthogonalization and $QR$–Decomposition

3.1 Gram-Schmidt Orthogonalization

Given linearly independent (the linearly independence assumption will be removed later) vectors $v_1, \ldots, v_k \in \mathbb{R}^n$, the Gram-Schmidt process constructs orthonormal vectors $q_1, \ldots, q_k$, i.e., vectors that satisfy

$$q_i^T q_j = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise}, \end{cases}$$

that span the same subspace as the vectors $v_1, \ldots, v_k$. More precisely, for $i = 1, \ldots, k$, the Gram-Schmidt process successively computes an orthonormal basis $\{q_1, \ldots, q_i\}$ from $\{v_1, \ldots, v_i\}$ such that both bases span the same subspace. The idea is to use orthogonal projection to remove components along the existing basis vectors, leaving an orthogonal set. The idea is illustrated in Figure 3.

![Figure 3](image.png)

Figure 3: The projectors $P_R(q) = q q^T / (q^T q)$ and $P_N(q^T) = I - P_R(q)$ decompose the vector $v$ into the component $P_R(q)v$ along $q$, and the orthogonal component $P_N(q^T)v$.

The steps of the Gram-Schmidt process are described next.

$i = 1$. If $i = 1$, then we need to find a vector $q_1$ with $q_1^T q_1 = 1$ such that

$$\text{span}\{q_1\} = \text{span}\{v_1\}.$$ 

The vector $q_1$ is obtained by normalizing $v_1$:

$$q_1 = v_1 / \|v_1\|_2.$$ 

$i = 2$. Given $q_1$ we want to compute $q_2$ such that $q_1^T q_2 = 0$, $q_2^T q_2 = 1$, and

$$\text{span}\{q_1, q_2\} = \text{span}\{v_1, v_2\}.$$ 

First note that since $\text{span}\{q_1\} = \text{span}\{v_1\}$ we have $\text{span}\{v_1, v_2\} = \text{span}\{q_1, v_2\}$. Moreover, by the Fundamental Theorem of Linear Algebra we can write $v_2$ as the sum of vector in
3.1 Gram-Schmidt Orthogonalization

\[ \text{span}\{q_1\} = \mathcal{R}(Q_1), \] where \( Q_1 \) is the matrix \( Q_1 = (q_1) \in \mathbb{R}^{n \times 1} \), and a vector in the orthogonal complement of \( \mathcal{R}(Q_1) \). We can use projections to express these vectors. The projection onto \( \mathcal{R}(Q_1) \) is given by \( P_{\mathcal{R}(Q_1)} = Q_1(Q_1^T Q_1)^{-1} Q_1^T \). Since \( q_1^T q_1 = 1 \), \( Q_1^T Q_1 = 1 \) and the projection is given by

\[ P_{\mathcal{R}(Q_1)} = Q_1 Q_1^T. \]

Hence,

\[ v_2 = Q_1 Q_1^T v_2 + (I - Q_1 Q_1^T) v_2. \]

Since \( Q_1 Q_1^T v_2 \in \mathcal{R}(Q_1) = \text{span}\{q_1\} \) we have

\[ \text{span}\{q_1, v_2\} = \text{span}\{q_1, (I - Q_1 Q_1^T) v_2\}. \]

The vector

\[ \tilde{q}_2 = (I - Q_1 Q_1^T) v_2 = v_2 - (q_1^T v_2) q_1 \]

is orthogonal to \( q_1 \). We just need to normalize it to obtain

\[ q_2 = \tilde{q}_2 / \|	ilde{q}_2\|_2. \]

We have

\[ \text{span}\{v_1, v_2\} = \text{span}\{q_1, v_2\} = \text{span}\{q_1, (I - Q_1 Q_1^T) v_2\} = \text{span}\{q_1, q_2\}. \]

\( i > 2 \). Given \( q_1, \ldots, q_{i-1} \) with

\[ q_l^T q_j = \begin{cases} 1 & \text{if } l = j, \\ 0 & \text{otherwise} \end{cases} \]

and \( \text{span}\{q_1, \ldots, q_{i-1}\} = \text{span}\{v_1, \ldots, v_{i-1}\} \), we want to compute \( q_i \) such that \( q_j^T q_i = 0 \), \( j = 1, \ldots, k - 1 \), \( q_i^T q_i = 1 \), and

\[ \text{span}\{q_1, \ldots, q_{i-1}, q_i\} = \text{span}\{v_1, \ldots, v_{i-1}, v_i\}. \]

Since \( \text{span}\{q_1, \ldots, q_{i-1}\} = \text{span}\{v_1, \ldots, v_{i-1}\} \) we have

\[ \text{span}\{v_1, \ldots, v_{i-1}, v_i\} = \text{span}\{q_1, \ldots, q_{i-1}, v_i\}. \]

By the Fundamental Theorem of Linear Algebra we can write \( v_i \) as the sum of vector in \( \text{span}\{q_1, \ldots, q_{i-1}\} = \mathcal{R}(Q_{i-1}) \), where \( Q_{i-1} \) is the matrix \( Q_{i-1} = (q_1, \ldots, q_{i-1}) \in \mathbb{R}^{n \times k-1} \), and a vector in the orthogonal complement of \( \mathcal{R}(Q_{i-1}) \). We can use projections to express these vectors. The projection onto \( \mathcal{R}(Q_{i-1}) \) is given by \( P_{\mathcal{R}(Q_{i-1})} = Q_{i-1}(Q_{i-1}^T Q_{i-1})^{-1} Q_{i-1}^T \). Since the columns \( q_1, \ldots, q_{i-1} \) of \( Q_{i-1} \) are orthonormal, \( Q_{i-1}^T Q_{i-1} = I \) and the projection is given by

\[ P_{\mathcal{R}(Q_{i-1})} = Q_{i-1} Q_{i-1}^T. \]
Hence,
\[ v_i = Q_{i-1}Q_{i-1}^T v_i + (I - Q_{i-1}Q_{i-1}^T) v_i. \]

Since \( Q_1Q_1^T v_2 \in \mathcal{R}(Q_1) = \text{span}\{q_1\} \) we have
\[ \text{span}\{q_1, \ldots, q_{i-1}, v_2\} = \text{span}\{q_1, \ldots, q_{i-1}, (I - Q_{i-1}Q_{i-1}^T) v_i\}. \]

The vector
\[ \tilde{q}_i = (I - Q_{i-1}Q_{i-1}^T)v_i = v_i - \sum_{j=1}^{i-1} (q_j^T v_i) q_j \]
is orthogonal to \( q_1, \ldots, q_{i-1} \). We just need to normalize it to obtain
\[ q_i = \tilde{q}_i / \|\tilde{q}_i\|_2. \]

We have
\[ \text{span}\{v_1, \ldots, v_{i-1}, v_i\} = \text{span}\{q_1, \ldots, q_{i-1}, v_i\} = \text{span}\{q_1, \ldots, q_{i-1}, (I - Q_{i-1}Q_{i-1}^T) v_i\} = \text{span}\{q_1, \ldots, q_{i-1}, q_i\}. \]

The above steps are summarized in the following algorithm.

---

**Algorithm 4 (Gram-Schmidt)**

0. Given linearly independent vectors \( v_1, \ldots, v_k \in \mathbb{R}^n \).

1. Set \( q_1 = v_1 / \|v_1\|_2. \)

2. For \( i = 1, \ldots, k - 1 \)
   a. Compute \( \tilde{q}_{i+1} = v_{i+1} - \sum_{j=1}^{i} q_j^T v_{i+1} q_j. \)
   b. Set \( q_{i+1} = \tilde{q}_{i+1} / \|\tilde{q}_{i+1}\|_2. \)

3. Return orthonormal vectors \( q_1, \ldots, q_k \in \mathbb{R}^n \) with
   \[ \text{span}\{v_1, \ldots, v_k\} = \text{span}\{q_1, \ldots, q_k\}. \]
Example 5  We apply the Gram-Schmidt Algorithm 4 to the vectors

\[ v_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad v_2 = \begin{pmatrix} 1 \\ 2 \\ 4 \\ 5 \end{pmatrix}, \quad v_3 = \begin{pmatrix} 1 \\ -3 \\ -4 \\ -2 \end{pmatrix}. \]

The Gram-Schmidt Algorithm 4 computes orthonormal vectors \( q_1, q_2, q_3 \) as follows:

\[
q_1 = \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix},
\]

\[
\tilde{q}_2 = \begin{pmatrix} 1 \\ 2 \\ 4 \\ 5 \end{pmatrix} - \frac{12}{4} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -2 \\ -1 \\ 1 \\ 2 \end{pmatrix}, \quad q_2 = \frac{1}{\sqrt{10}} \begin{pmatrix} -2 \\ -1 \\ 1 \\ 2 \end{pmatrix},
\]

\[
\tilde{q}_3 = \begin{pmatrix} 1 \\ -3 \\ -4 \\ -2 \end{pmatrix} - \frac{8}{4} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} - \frac{7}{10} \begin{pmatrix} -2 \\ -1 \\ 1 \\ 2 \end{pmatrix} = \frac{1}{10} \begin{pmatrix} 16 \\ -17 \\ -13 \\ 14 \end{pmatrix}, \quad q_3 = \frac{1}{\sqrt{910}} \begin{pmatrix} 16 \\ -17 \\ -13 \\ 14 \end{pmatrix}.
\]

So far we have assumed that the vectors \( v_1, \ldots, v_k \in \mathbb{R}^n \) are linearly independent. The next example explores what happens when they are not.
Example 6 We apply the Gram-Schmidt Algorithm to the vectors
\[ v_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad v_2 = \begin{pmatrix} 1 \\ 2 \\ 4 \\ 5 \end{pmatrix}, \quad v_3 = \begin{pmatrix} 0 \\ 1 \\ 3 \\ 4 \end{pmatrix}, \quad v_4 = \begin{pmatrix} 1 \\ -3 \\ -4 \\ -2 \end{pmatrix}. \]
Note that \( v_3 = v_2 - v_1 \), i.e. the vectors are linearly dependent.

The first two steps of the Gram-Schmidt Algorithm give
\[ q_1 = \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad \tilde{q}_2 = \left( \frac{2}{4} \right) - \frac{12}{4} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -2 \\ -1 \\ 1 \\ 2 \end{pmatrix}, \quad q_2 = \frac{1}{\sqrt{10}} \begin{pmatrix} -2 \\ -1 \\ 1 \\ 2 \end{pmatrix}. \]

The next steps leads to
\[ \tilde{q}_3 = \begin{pmatrix} 0 \\ 1 \\ 3 \\ 4 \end{pmatrix} - \frac{8}{4} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} - \frac{10}{10} \begin{pmatrix} -2 \\ -1 \\ 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \]

The vector \( \tilde{q}_3 \) is zero because \( v_3 \in \text{span}\{v_1, v_2\} = \text{span}\{q_1, q_2\} \). We define
\[ q_3 = \tilde{q}_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \]
and continue with the Gram-Schmidt orthogonalization. This gives
\[ \tilde{q}_4 = \begin{pmatrix} 1 \\ -3 \\ -4 \\ -2 \end{pmatrix} - \frac{8}{4} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} - \frac{7}{10} \begin{pmatrix} -2 \\ -1 \\ 1 \\ 2 \end{pmatrix} = \frac{1}{10} \begin{pmatrix} 16 \\ -17 \\ -13 \\ 14 \end{pmatrix}, \quad q_4 = \frac{1}{\sqrt{910}} \begin{pmatrix} 16 \\ -17 \\ -13 \\ 14 \end{pmatrix}. \]

In the end
\[ \text{span}\{v_1, v_2, v_3, v_4\} = \text{span}\{q_1, q_2, q_4\} = \text{span}\{q_j : q_j \neq 0, j \in \{1, \ldots, 4\}\}. \]
The generalization of the Gram-Schmidt Algorithm 4 to (possibly linearly dependent) vectors \( v_1, \ldots, v_i \in \mathbb{R}^n \) is now straightforward and is given in the following algorithm.

\[ \text{Algorithm 7 (Gram-Schmidt)} \]

0. Given vectors \( v_1, \ldots, v_i \in \mathbb{R}^n \) with \( v_1 \neq 0 \) (otherwise renumber).

1. Set \( q_1 = v_1 / \| v_1 \|_2 \).

2. For \( i = 1, \ldots, k - 1 \)
   
   a. Compute \( \tilde{q}_{i+1} = v_{i+1} - \sum_{j=1}^{i} q_j^T v_{i+1} q_j \).
   
   b. If \( \tilde{q}_{i+1} \neq 0 \), set \( q_{i+1} = \tilde{q}_{i+1} / \| \tilde{q}_{i+1} \|_2 \).
      
      If \( \tilde{q}_{i+1} = 0 \), set \( q_{i+1} = 0 \).

3. Return a set of orthonormal vectors \( \{ q_j : q_j \neq 0, j \in \{1, \ldots, k\} \} \) with \( \text{span}\{ v_1, \ldots, v_i \} = \text{span} \{ q_j : q_j \neq 0, j \in \{1, \ldots, k\} \} \).

3.2 \( QR \)-Decomposition via Gram-Schmidt

Assume that \( \tilde{q}_{i+1} \neq 0 \). The identity in step 2a of the Gram-Schmidt Algorithm 7 reads

\[ \| \tilde{q}_{i+1} \|_2 q_{i+1} = v_{i+1} - \sum_{j=1}^{i} q_j^T v_{i+1} q_j. \]  
(1)

Multiply by \( q_{i+1}^T \) to obtain

\[ \| \tilde{q}_{i+1} \|_2 \underbrace{q_{i+1}^T \tilde{q}_{i+1}}_{=1} = q_{i+1}^T v_{i+1} - \sum_{j=1}^{i} q_j^T v_{i+1} q_j. \]
Hence, $\|\tilde{q}_{i+1}\|_2 = q_{i+1}^Tv_{i+1}$. This identity and (1) give

$$v_{i+1} = \sum_{j=1}^{i+1} q_j^Tv_{i+1} = \begin{bmatrix} q_1, \ldots, q_{i+1} \end{bmatrix}_{n \times (i+1)} \begin{pmatrix} q_1^Tv_{i+1} \\ \vdots \\ q_{i+1}^Tv_{i+1} \end{pmatrix} = \begin{bmatrix} q_1, \ldots, q_{i+1}, q_{i+2}, \ldots, q_k \end{bmatrix}_{n \times k} \begin{pmatrix} q_1^Tv_{i+1} \\ \vdots \\ q_{i+1}^Tv_{i+1} \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \quad (2)$$

Let $\tilde{q}_{i+1} \neq 0$ for all $i = 0, \ldots, k - 1$. Equation (2) is the identity for the $(i + 1)$st column in

$$\begin{bmatrix} v_1, \ldots, v_{i+1}, v_{i+2}, \ldots, v_k \end{bmatrix} = V \in \mathbb{R}^{n \times k} = Q \in \mathbb{R}^{n \times k} = R \in \mathbb{R}^{k \times k}. \quad (3)$$

A decomposition

$$V = QR \quad (4)$$

where

$$Q \in \mathbb{R}^{n \times k} \quad \text{with} \quad Q^TQ = I$$

and

$$R \in \mathbb{R}^{k \times k} \quad \text{is upper triangular}$$

is called a $QR$-decomposition (of $V$).

**Example 8** Example 5 leads to the $QR$ decomposition

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & -3 \\ 1 & 4 & -4 \\ 1 & 5 & -2 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{\sqrt{10}}{2} \\ \frac{1}{2} & \frac{\sqrt{10}}{2} \\ \frac{1}{2} & \frac{\sqrt{10}}{2} \\ \frac{1}{2} & \frac{\sqrt{10}}{2} \end{pmatrix} \begin{pmatrix} \frac{1}{2} & \frac{\sqrt{10}}{2} & \frac{16}{\sqrt{910}} \\ 0 & \frac{\sqrt{10}}{2} & \frac{-17}{\sqrt{910}} \end{pmatrix} \begin{pmatrix} 2 & 6 & -4 \\ 0 & \sqrt{10} & \frac{-7}{\sqrt{910}} \\ 0 & 0 & \frac{91}{\sqrt{910}} \end{pmatrix}. \quad \diamond$$
So far we have assumed that \( \tilde{q}_{i+1} \neq 0 \) for all \( i = 0, \ldots, k - 1 \). If \( i \) is the first index such that \( \tilde{q}_{i+1} = 0 \), then the identity in step 2a of the Gram-Schmidt Algorithm 7 reads

\[
0 = v_{i+1} - \sum_{j=1}^{i} q_j^T v_{i+1} q_j,
\]

i.e.,

\[
v_{i+1} = \sum_{j=1}^{i} q_j^T v_{i+1} q_j = \begin{bmatrix} q_1, \ldots, q_i \end{bmatrix}_{n \times i} \begin{pmatrix} q_1^T v_{i+1} \\ \vdots \\ q_i^T v_{i+1} \end{pmatrix}.
\]

If \( \tilde{q}_{i+1} = 0 \) and all other \( \tilde{q}_{j+1} \neq 0 \) for all \( i = 0, \ldots, k - 1, j \neq i \), then

\[
v_{i+1} = \begin{bmatrix} q_1, \ldots, q_i \end{bmatrix}_{n \times i} \begin{pmatrix} q_1^T v_{i+1} \\ \vdots \\ q_{i+1}^T v_{i+1} \end{pmatrix} = \begin{bmatrix} q_1, \ldots, q_i, q_{i+2}, \ldots, q_k \end{bmatrix}_{n \times (k-1)} \begin{pmatrix} q_1^T v_{i+1} \\ \vdots \\ q_{i+1}^T v_{i+1} \\ 0 \\ \vdots \\ 0 \end{pmatrix}.
\]

We can now use (2) and (6) to decompose \( V \in \mathbb{R}^{n \times k} \) as

\[ V = QR \]

where now

\[ Q \in \mathbb{R}^{n \times (k-1)} \quad \text{with} \quad Q^T Q = I \]

and

\[ R \in \mathbb{R}^{(k-1) \times k} \quad \text{is upper triangular.} \]

More generally, if the Gram-Schmidt Algorithm 7 generates \( l \leq k \) nonzero vectors \( q_i \), then \( V \in \mathbb{R}^{n \times k} \) can be decomposed as

\[ V = QR \]

where now

\[ Q \in \mathbb{R}^{n \times l} \quad \text{with} \quad Q^T Q = I \]

and

\[ R \in \mathbb{R}^{l \times k} \quad \text{is upper triangular.} \]

**Example 9** Example 6 leads to the \( QR \) decomposition

\[
\begin{pmatrix}
1 & 1 & 0 & 1 \\
1 & 2 & 1 & -3 \\
1 & 4 & 3 & -4 \\
1 & 5 & 4 & -2
\end{pmatrix}
= \begin{pmatrix}
\frac{1}{2} & -\frac{2}{\sqrt{5}} & \frac{16}{\sqrt{910}} \\
\frac{1}{2} & \frac{1}{\sqrt{5}} & \frac{\sqrt{10}}{\sqrt{910}} \\
\frac{1}{2} & \frac{1}{\sqrt{10}} & \frac{\sqrt{10}}{\sqrt{910}} \\
\frac{1}{2} & \frac{2}{\sqrt{10}} & \frac{\sqrt{10}}{\sqrt{910}}
\end{pmatrix}
\begin{pmatrix}
2 & 6 & 4 & -4 \\
0 & \sqrt{10} & 0 & \frac{-7}{\sqrt{910}} \\
0 & 0 & 0 & \frac{-7}{\sqrt{910}}
\end{pmatrix}.
\]
3.3 Using the $QR$-Decomposition to Solve Linear Least Squares Problems

The crucial property for using the $QR$-decomposition to solve linear least squares problems, is the fact that an orthogonal matrix does not change the length (=2-norm) of a vector. Let

$$Q \in \mathbb{R}^{m \times n} \quad \text{with} \quad Q^T Q = I.$$  

If $z \in \mathbb{R}^n$ then

$$\|Qz\|_2^2 = (Qz)^T Qz = z^T Q^T Q z = z^T z = \|z\|_2^2,$$

i.e., $z$ and $Qz$ have the same norm (length). Note that $z \in \mathbb{R}^n$ and $Qz \in \mathbb{R}^m$ have different number of entries.

Now, let $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ and consider the least squares problem

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2. \quad (7)$$

We assume that $m \geq n$ (more measurements than unknowns) and we assume that $\text{rank}(A) = n$. We can use the QR decomposition for other cases as well.

We compute the $QR$-decomposition of $A$ (apply the Gram-Schmidt Algorithm 7 to the columns of $A$) to get

$$A = QR \quad (8)$$

where

$$Q \in \mathbb{R}^{m \times n} \quad \text{with} \quad Q^T Q = I_{n \times n}$$

and $\mathcal{R}(A) = \mathcal{R}(Q)$, and where

$$R \in \mathbb{R}^{n \times n} \quad \text{is upper triangular.}$$

We will see that the $QR$-decomposition of $A$, (8), allows us to transform the least squares problem (7) into a simpler one that can be solved directly.

First we write

$$b = QQ^T b + (I - QQ^T) b. \quad (9)$$

Since $Q$ has orthonormal columns, $QQ^T b$ is the projection of $b$ onto $\mathcal{R}(Q) = \mathcal{R}(A)$ and $(I - QQ^T) b$ is the projection of $b$ onto $\mathcal{R}(Q)^\perp = \mathcal{R}(A)^\perp$. Now we insert (8), (9) into (7).

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2 = \min_{x \in \mathbb{R}^n} \|Q(Rx - Q^T b) - (I - QQ^T) b\|_2^2.$$
Since $Q(Rx - Q^Tb) \in \mathcal{R}(Q)$ and $I - QQ^T)b$ are orthogonal,
\[
\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2 = \min_{x \in \mathbb{R}^n} \|Q(Rx - Q^Tb) - (I - QQ^T)b\|_2^2 \\
= \min_{x \in \mathbb{R}^n} \|Q(Rx - Q^Tb)\|_2^2 + \|(I - QQ^T)b\|_2^2 \\
= \min_{x \in \mathbb{R}^n} \|Rx - Q^Tb\|_2^2 + \|(I - QQ^T)b\|_2^2,
\]
where in the last equation we have used the property $\|Qz\|_2^2 = \|z\|_2^2$ for all $z \in \mathbb{R}^n$. Since $R \in \mathbb{R}^{n \times n}$ is upper triangular,
\[
Rx = Q^Tb
\]
can be solved by back substitution. If $x_*$ is the solution of (10), then $x_*$ solves the least squares problem (10) and
\[
\|Ax_* - b\|_2^2 = \min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2 = \|(I - QQ^T)b\|_2^2.
\]

**Example 10** Consider the linear least squares problem
\[
\min_{x \in \mathbb{R}^n} \|Ax - b\|_2
\]
with
\[
A = \begin{pmatrix}
1 & 1 & 1 \\
1 & 2 & -3 \\
1 & 4 & -4 \\
1 & 5 & -2
\end{pmatrix}, \quad b = \begin{pmatrix} 1 \\
2 \\
3 \\
4 \end{pmatrix}.
\]
See Example 8. The $QR$ decomposition of $A$ computed using Gram-Schmidt is (only first 4 digits shown)
\[
Q = \begin{pmatrix}
0.5000 & -0.6325 & 0.5304 \\
0.5000 & -0.3162 & -0.5635 \\
0.5000 & 0.3162 & -0.4309 \\
0.5000 & 0.6325 & 0.4641
\end{pmatrix}, \quad R = \begin{pmatrix}
2.0000 & 6.0000 & -4.0000 \\
0 & 3.1623 & -2.2136 \\
0 & 0 & 3.0166
\end{pmatrix}.
\]

We compute $Q^Tb$
\[
Q^Tb = \begin{pmatrix}
5.0000 \\
2.2136 \\
-0.0331
\end{pmatrix}.
\]
Solving the triangular linear system $Rx = Q^Tb$ yields the solution
\[
x = \begin{pmatrix}
0.4011 \\
0.6923 \\
-0.0110
\end{pmatrix}
\]
of the linear least squares problem.

---

CAAM 335 Fall 2016 updated October 13, 2016
3.4 QR–Decomposition in MATLAB

MATLAB’s QR–Decomposition

This section is about the QR decomposition computed by MATLAB and many other software packages. The QR decomposition of a matrix $V$ is not unique and a QR-decomposition can be computed with other methods than Gram-Schmidt. In fact, because only a small subset of the real numbers can be represented in a computer and therefore rounding takes place when computing, the Gram-Schmidt method can suffer severely from this rounding. Instead MATLAB uses so-called Householder transformations to compute the QR-decomposition. You will learn about those in a course on Numerical Analysis, such as CAAM 453. For a matrix $V \in \mathbb{R}^{n \times k}$ MATLAB computes a QR-decomposition

$$V = QR$$

where now

$$Q \in \mathbb{R}^{n \times n} \quad \text{with} \quad Q^T Q = I$$

(note $Q$ is square) and

$$R \in \mathbb{R}^{n \times k} \quad \text{is upper triangular.}$$

Example 11 We apply MATLAB’s qr to the matrix in Example 8.

```matlab
>> V = [1 1 1; 1 2 -3; 1 4 -4; 1 5 -2];
>> [Q,R] = qr(V)
```

$$Q =
\begin{bmatrix}
-0.5000 & -0.6325 & 0.5304 & -0.2621 \\
-0.5000 & -0.3162 & -0.5635 & 0.5766 \\
-0.5000 & 0.3162 & -0.4309 & -0.6814 \\
-0.5000 & 0.6325 & 0.4641 & 0.3669
\end{bmatrix}$$

$$R =
\begin{bmatrix}
-2.0000 & -6.0000 & 4.0000 \\
0 & 3.1623 & -2.2136 \\
0 & 0 & 3.0166 \\
0 & 0 & 0
\end{bmatrix}$$

Note that the first three columns in MATLAB’s $Q$ are $-q_1, q_2, q_3$ in Example 8, and the upper $3 \times 3$ submatrix in MATLAB’s $R$ is related to $R$ in Example 8.

Next, we apply MATLAB qr to the matrix in Example 9.

```matlab
>> V = [1 1 0 1; 1 2 1 -3; 1 4 3 -4; 1 5 4 -2];
>> [Q,R] = qr(V)
```

CAAM 335 Fall 2016 updated October 13, 2016
These matrices (their last two columns) are different than the matrices $Q$ and $R$
Example 9. 

In general the Matrices $Q$ and $R$ in MATLAB’s $QR$-decomposition are different from the one we compute by hand using the Gram-Schmidt method. However, the fundamental structure, and how we use either $QR$-decomposition is the same. We will discuss how to use the $QR$-decomposition to solve linear least squares problems.

**Using MATLAB’s $QR$-Decomposition to Solve Linear Least Squares Problems**

Let $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. Let $Q \in \mathbb{R}^{m \times m}$ be an orthogonal matrix and let $R \in \mathbb{R}^{n \times n}$ be an upper triangular matrix $R \in \mathbb{R}^{n \times n}$ such that

\[ A = Q \begin{pmatrix} R \\ 0 \end{pmatrix} \]

\[ \{n \} \{m-n \} \tag{12} \]

Note that since $Q \in \mathbb{R}^{m \times m}$ and $Q^T Q = I_{m \times m}$, $Q^{-1} = Q^T$. Note that $Q \in \mathbb{R}^{m \times m}$, not $Q \in \mathbb{R}^{m \times n}$. Since $Q^T Q = I$ we have that

\[ \|Q^T y\|_2 = \|y\|_2 \forall y \in \mathbb{R}^m. \]

With this observation and the decomposition (12) we obtain that

\[ \|Ax - b\|_2^2 = \|Q^T (Ax - b)\|_2^2 = \| \begin{pmatrix} R \\ 0 \end{pmatrix} x - Q^T b\|_2^2. \]

Now we partition $Q^T b$ as

\[ Q^T b = \begin{pmatrix} c \\ d \end{pmatrix} \]

\[ \{n \} \{m-n \} \tag{13} \]
This yields
\[
\|Ax - b\|_2^2 = \| \begin{pmatrix} R \\ 0 \end{pmatrix} y - \begin{pmatrix} c \\ d \end{pmatrix} \|_2^2 = \| \begin{pmatrix} Rx - c \\ -d \end{pmatrix} \|_2^2 \\
= \|Rx - c\|_2^2 + \|d\|_2^2.
\]

Using this transformation, the least squares problem can be rewritten as
\[
\min_x \|Ax - b\|_2^2 = \min_x \|Rx - c\|_2^2 + \|d\|_2^2.
\]

Now we observe that the term \(\|d\|_2^2\) does not depend on \(y\) and that \(\|Rx - c\|_2^2 \geq 0\) for all \(x\) and \(\|Rx_* - c\|_2^2 = 0\) if and only if \(x_*\) solves
\[
Rx = c. \tag{14}
\]

Thus,
\[
\|Ax - b\|_2^2 \geq \|d\|_2^2
\]
for all \(x\). Moreover, if \(x_*\) is the solution of \(14\), then
\[
\|Ax_* - b\|_2^2 = \|Rx_* - c\|_2^2 + \|d\|_2^2 = \|d\|_2^2.
\]

Hence,
\[
\|Ax_* - b\|_2^2 = \|d\|_2^2 = \min_x \|Ax - b\|_2^2,
\]
i.e. \(x_*\) is the solution of the linear least squares problem. Note that \(R\) is a nonsingular upper triangular matrix. Therefore \(14\) can be easily solved (backward solve).

**Algorithm 12** Compute the Solution of a Linear Least Squares Problem using the MATLAB QR–Decomposition

Let \(A \in \mathbb{R}^{m \times n}\) be a matrix with \(\text{rank}(A) = n\) and \(b \in \mathbb{R}^m\).

1. Compute an orthogonal matrix \(Q \in \mathbb{R}^{m \times m}\) and an upper triangular matrix \(R \in \mathbb{R}^{n \times n}\) such that
\[
Q^T A = \begin{pmatrix} R \\ 0 \end{pmatrix}.
\]

2. Compute
\[
Q^T b = \begin{pmatrix} c \\ d \end{pmatrix}.
\]

3. Solve \(Rx = c\).
Remark 13 If you use the MATLAB backslash command $A \backslash b$ to compute the solution of the linear least squares problem $\min \frac{1}{2} ||Ax - b||_2^2$, where $A \in \mathbb{R}^{m \times n}$, $m > n$, has rank($A$)=n, then MATLAB essentially uses Algorithm 12.

Example 14 Consider the linear least squares problem

$$\min ||Ax - b||_2$$

with

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & -3 \\ 1 & 4 & -4 \\ 1 & 5 & -2 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}.$$  

See Example 10. Using MATLAB’s qr gives (only first 4 digits shown)

$$Q = \begin{pmatrix} -0.5000 & -0.6325 & 0.5304 & -0.2621 \\ -0.5000 & -0.3162 & -0.5635 & 0.5766 \\ -0.5000 & 0.3162 & -0.4309 & -0.6814 \\ -0.5000 & 0.6325 & 0.4641 & 0.3669 \end{pmatrix}, \quad R = \begin{pmatrix} -2.0000 & -6.0000 & 4.0000 \\ 0 & 3.1623 & -2.2136 \\ 0 & 0 & 3.0166 \\ 0 & 0 & 0 \end{pmatrix}.$$  

We compute $Q^T b$ and determine $d, c$:

$$Q^T b = \begin{pmatrix} -5.0000 \\ 2.2136 \\ -0.0331 \\ 0.3145 \end{pmatrix}, \quad c = \begin{pmatrix} -5.0000 \\ 2.2136 \\ -0.0331 \end{pmatrix}, \quad d = (0.3145).$$  

Solving the triangular linear system $Rx = d$ yields the solution

$$x = \begin{pmatrix} 0.4011 \\ 0.6923 \\ -0.0110 \end{pmatrix}$$

of the linear least squares problem. The corresponding MATLAB code is given below.

```
>> A = [1 1 1; 1 2 -3; 1 4 -4; 1 5 -2];
>> b = [1; 2; 3; 4];
>> [Q,R] = qr(A)
```

$Q = 

-0.5000  -0.6325  0.5304  -0.2621 
-0.5000  -0.3162 -0.5635  0.5766 
-0.5000  0.3162 -0.4309 -0.6814 
-0.5000  0.6325  0.4641  0.3669

CAAM 335 Fall 2016 updated October 13, 2016
### 3.4 QR–Decomposition in MATLAB

\[
\begin{bmatrix}
-0.5000 & 0.3162 & -0.4309 & -0.6814 \\
-0.5000 & 0.6325 & 0.4641 & 0.3669 \\
\end{bmatrix}
\]

\[
R =
\begin{bmatrix}
-2.0000 & -6.0000 & 4.0000 \\
0 & 3.1623 & -2.2136 \\
0 & 0 & 3.0166 \\
0 & 0 & 0 \\
\end{bmatrix}
\]

\[
\text{>> cd = } Q' * b \\
\text{cd =}
\begin{bmatrix}
-5.0000 \\
2.2136 \\
-0.0331 \\
0.3145 \\
\end{bmatrix}
\]

\[
\text{>> x = } R(1:3,1:3) \backslash \text{cd}(1:3) \\
x =
\begin{bmatrix}
0.4011 \\
0.6923 \\
-0.0110 \\
\end{bmatrix}
\]