1. Let \( \mathbf{u} \in \mathbb{R}^n, \mathbf{u} \neq 0 \). A Householder matrix \( H_\mathbf{u} \in \mathbb{R}^{n \times n} \) is defined by
\[
H_\mathbf{u} = I - \frac{2\mathbf{u}\mathbf{u}^T}{\|\mathbf{u}\|_2^2}.
\]
(a) Show that \( H_\mathbf{u} \) is both symmetric and orthogonal.
(b) Show that for any \( \alpha \in \mathbb{R}, \alpha \neq 0 \),
\[
H_{\alpha\mathbf{u}} = H_\mathbf{u}.
\]
In other words, \( H_\mathbf{u} \) only depends on the ‘direction’ of \( \mathbf{u} \) and not on its ‘magnitude’.
(c) In general, given a matrix \( M \in \mathbb{R}^{n \times n} \) and a vector \( \mathbf{x} \in \mathbb{R}^n \), computing the matrix-vector product \( M\mathbf{x} \) requires \( n \) inner products — one for each row of \( M \) with \( \mathbf{x} \). Show that \( H_\mathbf{u}\mathbf{x} \) can be computed using only two inner products.
(d) Given \( \mathbf{a}, \mathbf{b} \in \mathbb{R}^n \) where \( \mathbf{a} \neq \mathbf{b} \) and \( \|\mathbf{a}\|_2 = \|\mathbf{b}\|_2 \). Find \( \mathbf{u} \in \mathbb{R}^n , \mathbf{u} \neq 0 \) such that
\[
H_\mathbf{u}\mathbf{a} = \mathbf{b}.
\]
(e) Show that \( \mathbf{u} \) is an eigenvector of \( H_\mathbf{u} \). What is the corresponding eigenvalue?
(f) Show that every \( \mathbf{v} \in \text{span}\{\mathbf{u}\}^\perp \) (cf. orthogonal complement in Homework 1) is an eigenvector of \( H_\mathbf{u} \). What are the corresponding eigenvalues? What is \( \dim(\text{span}\{\mathbf{u}\}^\perp) \)?
(g) Find the eigenvalue decomposition of \( H_\mathbf{u} \), i.e., find an orthogonal matrix \( Q \) and a diagonal matrix \( \Lambda \) such that
\[
H_\mathbf{u} = Q\Lambda Q^T.
\]

2. Let \( A \in \mathbb{R}^{m \times n} \) and suppose its complete orthogonal decomposition is given by
\[
A = Q_1 \begin{bmatrix} L & 0 \\ 0 & 0 \end{bmatrix} Q_2^T,
\]
where \( Q_1 \) and \( Q_2 \) are orthogonal, and \( L \) is an nonsingular lower triangular matrix. Recall that \( X \in \mathbb{R}^{n \times m} \) is the unique pseudo-inverse of \( A \) if the following Moore–Penrose conditions hold:
(i) \( AXA = A \),
(ii) \( XAX = X \),
(iii) \( (AX)^\dagger = AX \),
(iv) \( (XA)^\dagger =XA \)
and in which case we write \( X = A^\dagger \).
(a) Let
\[
A^\perp = Q_2 \begin{bmatrix} L^{-1} & Y \\ 0 & 0 \end{bmatrix} Q_1^T, \quad Y \neq 0.
\]
Which of the four conditions (i)–(iv) are satisfied?
(b) Prove that
\[
A^\dagger = Q_2 \begin{bmatrix} L^{-1} & 0 \\ 0 & 0 \end{bmatrix} Q_1^T
\]
by letting
\[ A^\dagger = Q_2 \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} Q_1^T \]
and by completing the following steps:
- Using (i), prove that \( X_{11} = L^{-1} \).
- Using the symmetry conditions (iii) and (iv), prove that \( X_{12} = 0 \) and \( X_{21} = 0 \).
- Using (ii), prove that \( X_{22} = 0 \).

3. Let \( A \in \mathbb{R}^{m \times n} \), \( b \in \mathbb{R}^m \), and \( c \in \mathbb{R}^n \). We are interested in the least squares problem
\[
\min_{x \in \mathbb{R}^n} \| Ax - b \|_2^2. \tag{3.1}
\]
(a) Show that \( x \) is a solution to (3.1) if and only if \( x \) is a solution to the augmented system
\[
\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}. \tag{3.2}
\]
(b) Show that the \((m + n) \times (m + n)\) matrix in (3.2) is nonsingular if and only if \( A \) has full column rank.
(c) Suppose \( A \) has full column rank and the QR decomposition of \( A \) is
\[ A = Q \begin{bmatrix} R \\ 0 \end{bmatrix}. \]
Show that the solution to the augmented system
\[
\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} b \\ c \end{bmatrix}
\]
can be computed from
\[ z = R^{-1}c, \quad \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = Q'b, \]
and
\[ x = R^{-1}(d_1 - z), \quad y = Q \begin{bmatrix} z \\ d_2 \end{bmatrix}. \]
(d) Hence deduce that if \( A \) has full column rank, then
\[ A^\dagger = R^{-1}Q_1^T \]
where \( Q = [Q_1, Q_2] \) with \( Q_1 \in \mathbb{R}^{m \times n} \) and \( Q_2 \in \mathbb{R}^{m \times (m-n)} \). Check that this agrees with the general formula derived for a rank-retaining factorization \( A = GH \) in the lectures.

4. Let \( A \in \mathbb{R}^{m \times n} \). Suppose we apply QR with column pivoting to obtain the decomposition
\[ A = Q \begin{bmatrix} R & S \\ 0 & 0 \end{bmatrix} \Pi^T \]
where \( Q \) is orthogonal and \( R \) is upper triangular and invertible. Let \( x_B \) be the basic solution, i.e.,
\[ x_B = \Pi \begin{bmatrix} R^{-1} \\ 0 \\ 0 \end{bmatrix} Q'b, \]
and let \( \hat{x} = A^\dagger b \). Show that
\[ \frac{\| x_B - \hat{x} \|_2}{\| x_B \|_2} \leq \| R^{-1}S \|_2. \]
(Hint: If
\[ \Pi^T \begin{bmatrix} u \\ v \end{bmatrix} \quad \text{and} \quad Q'b = \begin{bmatrix} c \\ d \end{bmatrix}, \]
consider the associated linearly constrained least-squares problem
\[
\min \|u\|_2^2 + \|v\|_2^2 \quad \text{s.t.} \quad Ru + Sv = c
\]
and write down the augmented system for the constrained problem.)

5. Given a symmetric \( A \in \mathbb{R}^{n \times n} \), \( 0 \neq x \in \mathbb{R}^n \), and \( b \in \mathbb{R}^n \). Let
\[
r = b - Ax
\]
Consider the QR decomposition
\[
[x, r] = QR
\]
and observe that if \( Ex = r \), then
\[
(Q^T E Q)(Q^T x) = Q^T r.
\]
Show how to compute a symmetric \( E \in \mathbb{R}^{n \times n} \) so that it attains
\[
\min_{(A+E)x=b} \|E\|_F.
\]

6. In this exercise, we will implement and compare Gram–Schmidt and Householder QR. Your implementation should be tailored to the program you are using for efficiency (e.g. vectorize your code in Matlab/Octave/Scilab). Assume in the following that the input is a matrix \( A \in \mathbb{R}^{m \times n} \) with \( \text{rank}(A) = n \leq m \) and we want to find its full QR decomposition \( A = QR \) where \( Q \in O(m) \) and \( R \in \mathbb{R}^{m \times n} \) is upper-triangular.

(a) Implement the (classical) Gram–Schmidt algorithm to obtain \( Q \) and \( R \).
(b) Implement the Householder QR algorithm to obtain \( Q \) and \( R \). You should (i) store \( Q \) implicitly, taking advantage of the fact that it can be uniquely specified by a sequence of vectors of decreasing dimensions; (ii) choose \( \alpha \) in your Householder matrices to have the opposite sign of \( x_1 \) to avoid cancellation in \( v_1 \) (cf. notations in lecture notes).
(c) Implement an algorithm for forming the product \( Qx \) and another for forming the product \( Q^T y \) when \( Q \) is stored implicitly as in (b).
(d) For increasing values of \( n \), generate an upper triangular \( R \in \mathbb{R}^{n \times n} \) and a \( B \in \mathbb{R}^{n \times n} \), both with random standard normal entries. Use your program’s built-in function for QR factorization to obtain a random\(^1 \) \( Q \in O(n) \) from the QR factorization of \( B \). Now form \( A = QR \) and apply your algorithms in (a) and (b) to find the QR factors of \( A \) — let these be \( \hat{Q} \) and \( \hat{R} \). Tabulate (using graphs with appropriate scales) the relative errors
\[
\frac{\|R - \hat{R}\|_F}{\|R\|_F}, \quad \frac{\|Q - \hat{Q}\|_F}{\|Q\|_F}, \quad \frac{\|A - \hat{Q}\hat{R}\|_F}{\|A\|_F},
\]
for various values of \( n \) and for each method. Scale \( Q, R, \hat{Q}, \hat{R} \) appropriately so that \( R \) and \( \hat{R} \) have positive diagonal elements.

(i) Comment on the relative errors in \( \hat{Q} \) and \( \hat{R} \) (these are called forward errors) versus the relative error in \( \hat{Q}\hat{R} \) (this is called backward error).
(ii) Comment on the relative error in \( \hat{Q}\hat{R} \) computed with Gram–Schmidt versus that computed with Householder QR.

\(^1\)This is usually how one would generate a random orthogonal matrix.
(e) Generate a Vandermonde matrix and a vector,

\[
A = \begin{bmatrix}
1 & \alpha_0 & \alpha_0^2 & \ldots & \alpha_0^{n-1} \\
1 & \alpha_1 & \alpha_1^2 & \ldots & \alpha_1^{n-1} \\
1 & \alpha_2 & \alpha_2^2 & \ldots & \alpha_2^{n-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \alpha_{m-1} & \alpha_{m-1}^2 & \ldots & \alpha_{m-1}^{n-1}
\end{bmatrix} \in \mathbb{R}^{m \times n}, \quad b = \begin{bmatrix}
\exp(\sin 4\alpha_0) \\
\exp(\sin 4\alpha_1) \\
\exp(\sin 4\alpha_2) \\
\vdots \\
\exp(\sin 4\alpha_{m-1})
\end{bmatrix} \in \mathbb{R}^m,
\]

where \( \alpha_i = i/(m - 1) \), \( i = 0, 1, \ldots, m - 1 \). This arises when we try to do polynomial fitting

\[
e^{\sin 4x} \approx c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1} x^{n-1}
\]

over the interval \([0, 1]\) at discrete points \( x = 0, \frac{1}{m-1}, \frac{2}{m-1}, \ldots, \frac{m-2}{m-1}, 1 \). For \( n = 15 \) and \( m = 100 \), solve the least squares problem \( \min \| Ax - b \|_2 \) and state your value of \( c_{14} \) using each of the following methods:

(i) Applying QR factorization to \( A \).

(ii) Applying QR factorization to the augmented matrix \([A, b] \in \mathbb{R}^{m \times (n+1)}\).

(iii) Solving the normal equations \( A^T A x = A^T b \).

For (i) and (ii), your code should show how the respective QR factors are used in obtaining a solution of the least squares problem. You are free to use your program’s built-in functions (e.g. \( \text{A\b} \) in Matlab/Octave/Scilab) for solving linear systems but for other things, use what you have implemented in (a), (b), (c). The true value of \( c_{14} \) is 2006.787453080206\ldots . Comment on the accuracy of each method and algorithm.