1. (a) Given 
\[ x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \neq 0 \]
find a $2 \times 2$ orthogonal matrix $H_\pm$ such that 

\[ H_\pm x = \pm \sqrt{x_1^2 + x_2^2} e_1. \]

(b) Given $x = x_j e_j + x_k e_k \neq 0$ where $j \neq k$, find an orthogonal matrix $H_\pm$ such that 

\[ H_\pm x = \pm \sqrt{x_j^2 + x_k^2} e_j. \]

(c) Given a tridiagonal matrix 
\[
A = \begin{bmatrix}
a_1 & c_1 & 0 & 0 & \cdots & 0 \\
b_2 & a_2 & c_2 & 0 & \cdots & 0 \\
0 & b_3 & a_3 & c_3 & \cdots & 0 \\
0 & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & \cdots & 0 & b_{n-1} & a_{n-1} & c_{n-1} \\
0 & \cdots & 0 & 0 & b_n & a_n
\end{bmatrix}
\]

with $n \geq 2$ even, find an orthogonal matrix $Q$ such that $QA$ has zeroes in place of even-numbered below-diagonal entries $b_{2j}$ for $j = 1 : n/2$.

(d) Use (c) to devise an algorithm for QR factorization of a tridiagonal matrix $A$.

Solution:
(a) We seek a rotation matrix $H_+$ that rotates a vector counterclockwise by $\theta$ that would have the following effect on a given nonzero $x$.

\[
\begin{pmatrix}
\cos(\theta) & -\sin(\theta) \\
\sin(\theta) & \cos(\theta)
\end{pmatrix}
\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \sqrt{x_1^2 + x_2^2} \\ 0 \end{pmatrix}
\]

From the second equation in the above matrix-vector product and using the fact that $\sin^2(\theta) + \cos^2(\theta) = 1$, we have:

\[
x_1 \sin(\theta) = -x_2 \cos(\theta) \implies x_1^2 \sin^2(\theta) = x_2^2 \cos^2(\theta) \\
\implies x_1^2 (1 - \cos^2(\theta)) - x_2^2 \cos^2(\theta) = 0 \\
\implies \cos^2(\theta) (x_1^2 + x_2^2) = x_1^2 \\
\implies \cos(\theta) = \frac{x_1}{\sqrt{x_1^2 + x_2^2}} \sin(\theta) = -\frac{x_2}{\sqrt{x_1^2 + x_2^2}}
\]

So the matrix $H_\pm$ we are looking for is

\[
H_\pm = \pm \begin{pmatrix}
\cos(\theta) & -\sin(\theta) \\
\sin(\theta) & \cos(\theta)
\end{pmatrix}
= \pm \begin{pmatrix}
\frac{x_1}{\sqrt{x_1^2 + x_2^2}} & \frac{x_2}{\sqrt{x_1^2 + x_2^2}} \\
\frac{x_2}{\sqrt{x_1^2 + x_2^2}} & \frac{x_1}{\sqrt{x_1^2 + x_2^2}}
\end{pmatrix}
\]

(b) The matrix $H_\pm$ has the form:
\[ H_{\pm} = \pm \begin{pmatrix}
1 & & & & & \\
1 & \cos(\theta) & -\sin(\theta) & & & \\
& \cdots & \ddots & \ddots & & \\
& & \sin(\theta) & \cos(\theta) & & \\
& & & \cdots & 1 & \\
& & & & 1 & \\
\end{pmatrix} \begin{pmatrix}
j \\
k \\
\end{pmatrix} \]

(c) We want to apply multiple Givens Rotations to zero out the entries \( b_2, b_4, b_6, \ldots, b_n \) in order starting from the leftmost column of the original matrix \( A \) and going to the right.

If we denote the matrix from part (b) to be \( H_{jk} \), then we have that

\[ Q = H_{n-1,n} \cdots H_{34} H_{12} \]

In fact, if we explicitly write out the individual Givens rotations as such:

\[ H_{12} = \begin{pmatrix}
\cos(\theta_1) & -\sin(\theta_1) \\
\sin(\theta_1) & \cos(\theta_1) \\
1 & & \cdots & \ddots & & \\
& \cdots & \ddots & \ddots & \ddots & & \\
& & & \cdots & 1 & \\
& & & & 1 & \\
\end{pmatrix} \]

\[ H_{34} = \begin{pmatrix}
1 & & & & & \\
1 & \cos(\theta_2) & -\sin(\theta_2) & & & \\
& \cdots & \ddots & \ddots & & \\
& & \sin(\theta_2) & \cos(\theta_2) & & \\
& & & \cdots & 1 & \\
& & & & 1 & \\
\end{pmatrix} \]

\[ H_{n-1,n} = \begin{pmatrix}
1 & & & & & \\
1 & & & & & \\
& \cdots & \ddots & \ddots & & \\
& & & \cdots & 1 & \\
& & & & \cos(\theta_{n/2}) & -\sin(\theta_{n/2}) \\
& & & & \sin(\theta_{n/2}) & \cos(\theta_{n/2}) \\
\end{pmatrix} \]

We can see that their product \( Q \) has the nice sparsity pattern of:

\[ Q = \begin{pmatrix}
\cos(\theta_1) & -\sin(\theta_1) & & & \\
\sin(\theta_1) & \cos(\theta_1) & & & \\
& \cos(\theta_2) & -\sin(\theta_2) & & \\
& \sin(\theta_2) & \cos(\theta_2) & & \\
& & \cdots & \ddots & \ddots & & \\
& & & \cdots & 1 & \\
& & & & \cos(\theta_{n/2}) & -\sin(\theta_{n/2}) \\
& & & & \sin(\theta_{n/2}) & \cos(\theta_{n/2}) \\
\end{pmatrix} \]
(d) This is only a modest extension of part (c). Now we need to zero out $b_2, b_3, b_4, \ldots, b_{n-1}, b_n$ in order so we apply Givens rotations one by one by left multiplication of $A$. So using the notation of the previous problem we have that

$$Q^T = H_{n-1,n} H_{n-2,n-1} \cdots H_3 H_2 H_{12}$$

where $Q^T A = R \implies A = QR$.

2. Use the pseudoinverse to study high-order quadrature formulas with arbitrary points, as follows. Write a function which accepts an integer $m > 0$, an integer $n \geq m$, $n+1$ arbitrary points $x_i$ and an interval $[a, b]$ and computes $n+1$ weights $w_i$ for a quadrature formula

$$\int_a^b f(x)dx = \sum_{i=0}^n w_i f(x_i).$$

The weights are to be computed as the minimum 2-norm solution of the underdetermined linear system which expresses the requirement that the formula be exact for $f(x) = 1, x, \ldots, x^m$ where $m$ may not be equal to $n$. Test your function with random, equispaced and Chebyshev points, on the interval $[0, 1]$, for $n = 2, 4, 8, 16, 32, 64, 128$ and $m$ varying from $n/2$ to $n$. In each case, compute the maximum error in integrating $1, x, \ldots, x^m$, and $f(x) = \cos(10x)$, and compute the quantity $\kappa = \sum_{i=0}^n |w_i|$. Draw general conclusions about formulas of high order $m$ with $n$ arbitrary points.

**Solution:** See the attached code PS4prob2.m. All variable names in text refer to the ones generated by PS4prob2.m.

Below is a plot for the absolute error in integration for $f(x) = \cos(10x)$ with $n = 32$. Full data for all $n$ is stored in the variables errorequi, errorrand, errorcheb.

![Integration error for $f(x) = \cos(10x)$](image)

We note that for all three choices of input points $\sum_{i=0}^n w_i = 1$, which reflects that fact that constants are always integrated exactly. So $\kappa$ would give a measure of how far the weights
deviate from being positive. \( \kappa = \frac{1}{|b-a|} \sum_{i=0}^{n} |w_i| \) is the condition number of the quadrature rule in the sense that the quadrature error is bounded by \( \kappa \) times the best approximation error. So a good rule need not have positive weights, but can get by with a few slightly negative ones. Below is a plot for \( n = 32 \). We note that in all cases, for a given \( n \), as \( m \) varies from \( n/2 \) to \( n \).

- Random case: \( \kappa \neq 1 \) already for small \( m \) and blows up dramatically with \( m \).
- Equispaced case: \( \kappa = 1 \) up until about \( m = n/2 \), but then \( \kappa \) starts to grow. The larger \( n \) is, the less dramatic the growth as \( m \) reaches \( n \).
- Chebyshev case: \( \kappa = 1 \) always.

Here are some general conclusions for high order \( m \) with \( n \) arbitrary points.

- Uniform nodes result in increasingly oscillatory and ill conditioned weights for high order quadrature rules. Chebyshev nodes fix this problem. This is analogous to what we observe in interpolation where Chebyshev nodes reduce the Runge phenomenon so the best approximation error tends to be smaller.

- The benefit of using Chebyshev nodes is reflected in the fact that they result in the most accurate quadrature rules.

3 Repeat the calculation of problem (2) with the weights computed instead from the linear system which expresses the requirement that the first \( m \) Legendre polynomials \( P_k(x) \) (shifted and scaled to the interval \([a,b] \)) be integrated exactly. The Legendre polynomials on \([a,b] \) can be evaluated by the recurrence

\[
P_{k+1}(x) = ((2k+1)tP_k(x) - kP_{k-1}(x))/(k+1)
\]

where \( t = (x - c)/h \) with \( c = (a + b)/2 \) and \( h = (b - a)/2 \). Start with \( P_0 = 1 \) and \( P_1(x) = t \).

**Solution:** The main thing to note here is that this results in more accurate high order quadrature rules, which is most apparent in the rule derived from the Chebyshev nodes.