3. QR-decomposition

1. Write an M-file for QR-decomposition using Gram-Schmidt ortogonalisation. Let us call the function to: `gramschmidt`

   - Input parameter: a square matrix (A)
   - Output arguments: an orthogonal matrix (Q) and an upper triangular matrix (R), such that satisfy $A = Q \cdot R$
   - To check existence of decomposition (the columns of A have to be linear independent) we can use any included function of Matlab.
   - The included functions can be used for computing norms, but we can compute via definition also.

2. Write an M-file to give the matrix of a Householder transformation, from a known point and its image. The name of function let be: `householder`

   - Input parameters: the coordinatas of the point and its image (P, P') Point P (and ofcourse P' also) can be from $\mathbb{R}^n$ where n is not predetermined.
   - Output argument: the matrix of Householder-transformation
   - Take care of choosing sign during transformation (the parameter σ effects the stability of the method)

3. The third function will asking data via graphical input. (It works for 2D points) Display points and the hyperspace of reflection. Ask for another point (also via graphical input) and apply the transformation to the new point. The function `householder` can be called during the algorithm.

   the name of function: `hhgraph`

4. Write an M-filet to realize QR-decomposition with Householder algorithm. Let us call our function to: `hhalg`

   - Input parameter: a square matrix (A)
   - Output arguments: an orthogonal matrix (Q) and an upper triangular matrix (R), such that satisfy $A = Q \cdot R$
   - The previous functions can be called.