1. Write a Matlab function \([\mathbf{Q}, \mathbf{R}] = \text{mgs}(\mathbf{A})\) (see the discussion in the classnotes of stability of the Gram-Schmidt algorithms) that computes a reduced QR factorization \(\mathbf{A} = \mathbf{QR}\) of an \(m \times n\) matrix \(\mathbf{A}\) with \(m \geq n\) using modified Gram-Schmidt orthogonalization. The output variables are a matrix \(\mathbf{Q} \in \mathbb{C}^{m \times n}\) with orthonormal columns and a triangular matrix \(\mathbf{R} \in \mathbb{C}^{n \times n}\).

2. (a) Write a Matlab program that sets up a 15 \times 40 matrix with entries 0 everywhere except for the values 1 in the positions indicated in the picture below. The upper-leftmost 1 is in position (2,2), and the lower-rightmost 1 is in position (13,39). This picture was produced with the command `spy(A)`.

```
0 5 10 15 20 25 30 35 40
0 5 10 15

HELLO
```

(b) Call `svd` to compute the singular values of \(\mathbf{A}\), and print the results. Plot these numbers using both `plot` and `semilogy`. What is the mathematically exact rank of \(\mathbf{A}\)? How does this show up in the computed singular values?

(c) For each \(i\) from 1 to rank(\(\mathbf{A}\)), construct the rank-\(i\) matrix \(\mathbf{B}\) that is the best approximation to \(\mathbf{A}\) in the 2-norm. Use the command `pcolor(B)` with `colormap(gray)` to create images of these various approximations.

3. (a) Write a Matlab function \([\mathbf{W}, \mathbf{R}] = \text{house}(\mathbf{A})\) that computes an implicit representation of a full QR factorization \(\mathbf{A} = \mathbf{QR}\) of an \(m \times n\) matrix \(\mathbf{A}\) with \(m \geq n\) using Householder reflections. The output variables are a lower-triangular matrix \(\mathbf{W} \in \mathbb{C}^{m \times n}\) whose columns are the vectors \(\mathbf{v}_k\) defining the successive Householder reflections, and a triangular matrix \(\mathbf{R} \in \mathbb{C}^{n \times n}\).

(b) Write a Matlab function \(\mathbf{Q} = \text{formQ}(\mathbf{W})\) that takes the matrix \(\mathbf{W}\) produced by `house` as input and generates a corresponding \(m \times m\) orthogonal matrix \(\mathbf{Q}\).

4. Let \(\mathbf{Z}\) be the matrix

\[
\mathbf{Z} = \begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 7 \\
4 & 2 & 3 \\
4 & 2 & 2
\end{bmatrix}.
\]

Compute the reduced QR factorization of \(\mathbf{Z}\) in Matlab: by the Gram-Schmidt routine `mgs` of Problem 1, by the Householder routines `house` and `formQ` of the previous problem, and by Matlab’s built-in command \([\mathbf{Q}, \mathbf{R}] = \text{qr}(\mathbf{Z}, 0)\). Compare these three and comment on any differences you see.