These questions are not designed to be easy or difficult, but to aid our understanding during class.

1. At your leisure, read Chapters 3, 4, and 5 of “Numerical Computing with IEEE Floating Point Arithmetic” by Michael Overton.

2. (Triangular orthogonalization) Find out about modified Gram–Schmidt. Show that modified Gram–Schmidt applied to an $n \times n$ real invertible matrix $A$ is equivalent to applying a sequence of upper-triangular matrices on the right of $A$ to obtain an orthogonal matrix, i.e., there are upper-triangular matrices $R_1, \ldots, R_n$ such that

$$AR_1 \cdots R_n = Q,$$

where $Q$ is an orthogonal matrix. $A = Q(R_n^{-1} \cdots R_1^{-1})$ is a QR factorization of $A$. (Please do not use this algorithm to compute $A = QR$, we have a more stable algorithm. See class.)

3. (Orthogonal triangularization) Read about Given’s rotations. Show that you can make an $n \times n$ real matrix $A$ upper-triangular by applying a sequence of orthogonal matrices on the left of $A$ to obtain a triangular matrix, i.e., there are orthogonal matrices $Q_1, \ldots, Q_n$ such that

$$(Q_1 \cdots Q_n)A = R,$$

where $R$ is an upper-triangular matrix. $A = (Q_n^T \cdots Q_1^T)R$ is a QR factorization of $A$. (Please do not use this algorithm to compute $A = QR$, we have a faster algorithm. See class.)

4. If $A$ is an invertible matrix, then how many ways are there to decompose $A$ into the factorization $A = QR$, where $Q$ is an orthogonal matrix and $R$ is upper-triangular?