Math 566 **Homework 3** **Due Feb. 17, 2023**
Grady Wright

- Problems marked NLA (Numerical Linear Algebra) are from the book.

1. (**Classical vs. modified Gram-Schmidt**, 20 pts) In this problem you will compare the QR decompositions computed using the classical (Algorithm 7.1 in NLA) and modified (Algorithm 8.1 in NLA) Gram-Schmidt algorithms. You will do these comparisons using the following matrix:

$$A = \begin{bmatrix} 1 & t_1 & t_1^2 & \cdots & t_1^{n-1} \\ 1 & t_2 & t_2^2 & \cdots & t_2^{n-1} \\ 1 & t_3 & t_3^2 & \cdots & t_3^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & t_m & t_m^2 & \cdots & t_m^{n-1} \end{bmatrix}, \tag{1}$$

where $t_j = (j-1)/(m-1)$ for $j = 1, \ldots, m$ (i.e. m equally spaced points over $[0,1]$). This is known as a Vandermonde matrix and arises in polynomial interpolation and least squares problems (however, there are other techniques that may work better than using the Vandermonde matrix in these problems).

   (a) Write a function that implements the classical Gram-Schmidt algorithm for computing the QR factorization of a m-by-n matrix A, where $m \geq n$. Turn in a listing of this code in your homework.

   (b) Repeat part (a), but now implement the modified Gram-Schmidt algorithm. Turn in a listing of this code in your homework.

   (c) Use your functions from part (a) and (b) to compute the QR factorization of A in (1) with $m = 100$, and $n = 15$. Report the $\|A - QR\|_\infty$ and $\|I - Q^T Q\|_\infty$ (I is the n-by-n identity matrix) for each code. Comment on the results. Do you find anything strange with results for these two norms?

2. (**Householder reflections**, 15 pts)

   (a) For any two real vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$ with the property $\mathbf{x}^T\mathbf{x} = \mathbf{y}^T\mathbf{y}$, there exists a Householder matrix H, with the property $H\mathbf{x} = \mathbf{y}$. Verify directly that if we let $\mathbf{v} = \dfrac{\mathbf{y} - \mathbf{x}}{\|\mathbf{y} - \mathbf{x}\|}$ and $H = I - 2\mathbf{v}\mathbf{v}^T$ then $H\mathbf{x} = (I - 2\mathbf{v}\mathbf{v}^T)\mathbf{x} = \mathbf{y}$.

   (b) Let H be a Householder matrix of size m for some real vectors $\mathbf{x}$ and $\mathbf{y}$ satisfying $\mathbf{x}^T\mathbf{x} = \mathbf{y}^T\mathbf{y}$. We showed in class that H is both symmetric and orthogonal. This means that the only possible eigenvalues of H are ± 1.

      i. Determine $\mathrm{Tr}(H)$ (i.e. the trace of H) and use this result together with the result from (a) to determine all the eigenvalues of H.

      ii. Show that $H\mathbf{v} = -\mathbf{v}$ and that $H\mathbf{u} = \mathbf{u}$ for any $\mathbf{u} \in \mathbb{R}^m$ that is orthogonal to $\mathbf{v}$. This gives all the eigenvectors of H. Now use this result to also determine the eigenvalues of H.

      iii. Using the properties of eigenvalues, determine $\det(H)$.

3. (**QR decomposition via Householder reflections**, 20 pts)

(a) Write a function called `house`, using for example MATLAB, that computes the implicit representation of a full QR decomposition of a real m-by-n matrix A via Householder reflections (Algorithm 10.1 of NLA). The function should take as input a matrix $A \in \mathbb{R}^{m\times n}$ and return as output a lower triangular matrix $V \in \mathbb{R}^{m\times n}$ whose columns are the vectors v_k defining the k^{th} Householder reflection, $k = 1, \ldots, n$, and an upper triangular matrix $R \in \mathbb{R}^{n\times n}$. Turn in a listing of your function.

(b) Write a function called `house2q` that takes as input the matrix V from part (a) and computes the corresponding m-by-m orthogonal matrix Q. You can do this efficiently by applying Algorithm 10.3 of NLA to the columns of the identity matrix. Turn in a listing of your function.

(c) Test your code on the matrix

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 7 \\ 4 & 2 & 3 \\ 4 & 2 & 2 \end{bmatrix}$$

Compare the R and Q you get with the codes from part (a) and (b) with the a QR decomposition function that comes with the software you are using (e.g. `qr` in MATLAB, NumPy, or Julia). Ensure that your code produces a Q and R such that $A = QR$ (at least to machine precision).

You may not use any functions available in software libraries, such as the MATLAB, NumPy, or Julia `qr`, for parts (a) or (b). Additionally, your code for (a) and (b) should avoid unnecessary FLOPs such as multiplying an identity matrix times another matrix.