Homework 5.

Amath 352
Applied Linear Algebra and Numerical Analysis

Due: 2/10/23 at 11:59pm to Gradescope

Directions:

Complete all component skills exercises and the multi-step problem as neatly as possible. **Attach all code and its output to the end of your assignment.** Up to 2 points may be deducted for homework that is illegible and/or poorly organized. You are encouraged to type homework solutions, and a half bonus point will be awarded to students who use LaTeX. (Check out my LaTeX beginner document and overleaf.com if you are new to LaTeX.) If you prefer not to type homeworks, I ask that **homeworks be scanned.** (I will not accept physical copies.) In addition, **homeworks must be in .pdf format.**

Pro-Tips:

- You have access to some solutions of the textbook exercises and are encouraged to use them. Note that these solutions are not always correct, so double check your work just in case.
- Teamwork makes the dream work, but please indicate at the top of your assignment who your collaborators are.
- Don't wait until the last minute. ☺

Component Skills Exercises

Exercise 1. (CS3.3)

In lecture, we defined the dot product geometrically as follows:

$$\varphi(\mathbf{u}, \mathbf{v}) = ||\mathbf{u}||_2 ||\mathbf{v}||_2 \cos(\theta),$$

where θ is the angle between $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$. One can show that this definition of the dot product satisfies the following three properties:

- $\varphi(\mathbf{u}, \mathbf{u}) \geq 0$ and $\varphi(\mathbf{u}, \mathbf{u}) = 0$ iff $\mathbf{u} = 0$ for all $\mathbf{u} \in \mathbb{R}^n$,
- $\varphi(\mathbf{u}, \mathbf{v}) = \varphi(\mathbf{v}, \mathbf{u})$ for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, and
- $\varphi(\mathbf{u} + \alpha\mathbf{w}, \mathbf{v}) = \varphi(\mathbf{u}, \mathbf{v}) + \alpha\varphi(\mathbf{w}, \mathbf{v})$, where $\alpha \in \mathbb{R}$ and $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbb{R}^n$.

Assuming these three properties are true, answer the following about φ.

a. Let $\{\mathbf{e}_1, ..., \mathbf{e}_n\}$ represent the canonical basis vectors in $\mathbb{R}^n$. (These are the column vectors of $\mathbf{I}$, the $n \times n$ identity matrix.) Show that

$$\varphi(\mathbf{e}_i, \mathbf{e}_j) = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}.$$

Note: the symbol δ_{ij} is called the Kronecker delta.

b. Using any of the properties above, show that

$$\varphi(\mathbf{u}, \mathbf{v} + \alpha\mathbf{w}) = \varphi(\mathbf{u}, \mathbf{v}) + \alpha\varphi(\mathbf{u}, \mathbf{w})$$

for any $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$.

c. Let $\mathbf{u} = u_1\mathbf{e}_1 + \cdots + u_n\mathbf{e}_n \in \mathbb{R}^n$ and $\mathbf{v} = v_1\mathbf{e}_1 + \cdots + v_n\mathbf{e}_n \in \mathbb{R}^n$. Using your answers from (a)-(b) as well as the properties of φ above, show that

$$\varphi(\mathbf{u}, \mathbf{v}) = u_1v_1 + \cdots + u_nv_n.$$

d. Using your answer to (c), show that

$$\phi(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{v},$$

where T represents the matrix transpose. This is yet another way to express the dot product.

Exercise 2. (CS3.3)

Consider the vectors

$$\mathbf{u} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \quad \text{and} \quad \mathbf{v} = \begin{pmatrix} 3 \\ 3 \\ \sqrt{6} \end{pmatrix}.$$

a. Compute the 1-norm, 2-norm, and ∞-norm of $\mathbf{u}$ and $\mathbf{v}$.

b. Compute the angle between $\mathbf{u}$ and $\mathbf{v}$. Please express your angle exactly—no decimals!

c. Compute the projection of $\mathbf{v}$ onto $\mathbf{u}$.

Exercise 3. (CS3.3)

From Olver and Shakiban, complete Exercise 3.2.15(a).

Exercise 4. (CS3.4)

From Olver and Shakiban, complete the following exercises:

a. Exercise 4.2.1(c).

b. Exercise 4.2.3. This should be a one sentence answer supported by a small calculation.

c. Exercise 4.2.4(c). This set is described by a plane. Can you find the equation of the plane?

Exercise 5. (CS3.5)

From Olver and Shakiban, complete Exercise 4.3.27(c).

Multi-Step Problem

Note: This problem will involve programming. Please attach your code and its output to the end of your homework assignment.

Given a $n \times k$ matrix $\mathbf{A}$ that is **full rank**, the **classical Gram-Schmidt algorithm** applied to the columns of the matrix $\mathbf{A}$ can be coded in MATLAB as follows:

```
[n,k] = size(A);
Q = zeros(n,k); % Allocate space for the new orthonormal column vectors.

for j = 1:k % Construct Q matrix column by column.
   v_j = A(:,j);
   for p = 1:j-1 % Construct v_j perpendicular to q_j.
      v_j = v_j - (transpose(Q(:,p))*A(:,j))*Q(:,p);
   end
   if norm(v_j,2)==0  % Check if two norm of v_j is zero.
      error('Matrix A is not full rank.');
   end
   Q(:,j) = v_j/norm(v_j,2);
end
```

The columns of the matrix $\mathbf{Q}$ are the orthonormal basis vectors that span the columns of $\mathbf{A}$.

Remark: One can show (and you are welcome to show) that this algorithm requires $2k^2n + \frac{1}{2}k(2n - k - 1)$ flops. If the matrix $\mathbf{A}$ is $n \times n$ and large, the asymptotic flop count of classical Gram-Schmidt is $\sim 2n^3$. Thus, orthogonalization is slightly more expensive than Gaussian elimination.

a. Create a new script in MATLAB called hw4_msp.m and start a new section of your code. Define the innocent-looking matrix

$$\mathbf{A} = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{pmatrix}.$$

Remark: The matrix above is the 5×5 *Hilbert matrix.* These matrices appear when approximating functions by polynomials in the least-squares sense. The columns of a Hilbert matrix are close to being linearly dependent, which makes the matrix $\mathbf{A}$ difficult to orthogonalize by Gram-Schmidt, as we will soon see.

Apply the classical Gram-Schmidt code to the Hilbert matrix above. Output your $\mathbf{Q}$ matrix to the Command Window. Be sure to show all decimal places.

b. Recall for an orthogonal matrix that $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$. Thus, for your answer to (a), we should expect $\mathbf{Q}^T\mathbf{Q} - \mathbf{I}$ to be close to the zero matrix. Let's define the error $\mathcal{E}$ of the classical Gram-Schmidt algorithm as

$$\mathcal{E} = \max_{\substack{1 \le i \le n \\ 1 \le j \le n}} |\left(\mathbf{Q}^T\mathbf{Q} - \mathbf{I}\right)_{ij}|,$$

i.e., the error $\mathcal{E}$ is the maximum entry in absolute value of the matrix $\mathbf{Q}^T\mathbf{Q} - \mathbf{I}$. Display this error to the Command Window for the $\mathbf{Q}$ obtained in (a). Show all decimal places. Based on this error, roughly how many decimal places should you trust the $\mathbf{Q}$ obtained in (a)?

c. The **condition number** is a measure of how linearly independent/dependent the columns of a given matrix are. The larger the condition number, the closer the columns of the matrix are to being linearly dependent. The precise definition of the condition number requires us to know the singular value decomposition, which we will get to later in the course. To compute the condition number of a matrix $\mathbf{A}$ in MATLAB, use cond($\mathbf{A}$).

Given the condition number of a matrix cond($\mathbf{A}$), the general rule of thumb is that any numerical computations involving the matrix $\mathbf{A}$ will lose approximately

$$\kappa = \log_{10}(\text{cond}(\mathbf{A}))$$

number of significant digits. In a separate section of your code, compute the condition number of the Hilbert matrix above in MATLAB. Display the condition number to the Command Window. Be sure to show all decimal places. Roughly how many significant digits do you expect to lose when performing Gram-Schmidt on the Hilbert matrix above?