18.085/18.0851 Computational Mathematical Engineering I **Summer 2019**

Homework 1

Homework: Richard Zhang *Scribes: Richard Zhang*

1.1 Gaussian Elimination

Please get comfortable with Gaussian eliminations by performing the operation on the following matrices

$$A_1 = \begin{bmatrix} 1 & -3 & 2 & 1 \\ 2 & -4 & 0 & -1 \\ -1 & 0 & -2 & 3 \\ 3 & -3 & 0 & -2 \end{bmatrix}, A_2 = \begin{bmatrix} -2 & 0 & -2 \\ 8 & 10 & -4 \\ 0 & -4 & 3 \end{bmatrix} \tag{1.1}$$

Also solve for x in $A_2 x = b$, where $b = [1, 1, -2]^T$

1.2 Null Space and Column Space

Find the null space and the column space of the following matrices. Verify the rank-nullity theorem on B_1

$$B_1 = \begin{bmatrix} 0 & -1 & 3 \\ 1 & 4 & -1 \\ 0 & 0 & 0 \end{bmatrix}, B_2 = \begin{bmatrix} 1 & -3 & 2 & 1 \\ 2 & -4 & 0 & -1 \\ -1 & 0 & -2 & 3 \\ 3 & -3 & 0 & -2 \end{bmatrix} \tag{1.2}$$

Hint: since B_2 is the same as A_1 in the first problem, what does the Gaussian elimination tell you about the matrix? Can you figure out the basis of its null and column space right away using the theorems in the notes?

1.3 Eigenvalues and Eigenvectors

Find the eigenvalues and eigenvectors of the following matrix

$$C_1 = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \tag{1.3}$$

1.4 L^p Distance and Quadratic Form

We will be using two important concepts in our future studies. For now, let's practice the computations of the two quantities. First, here are two definitions

Definition 1.1 *We define the L^p distance of two vectors, $\vec{u} = (u_1, ..., u_n)^T$ and $\vec{v} = (v_1, ..., v_n)^T$ as*

$$||\vec{u} - \vec{v}||_{L^p} = \left(\sum_{k=1}^{n} |u_k - v_k|^p \right)^{1/p} \tag{1.4}$$

Definition 1.2 *For vectors $u \in \mathbb{R}^m$ and $v \in \mathbb{R}^n$, and matrix $A \in \mathbb{R}^{m \times n}$, we define the quadratic form, $Q(u, v)$ as*

$$Q(u, v) = u^T A v \tag{1.5}$$

Here is one matrix

$$A_2 = \begin{bmatrix} -2 & 0 \\ 0 & -4 \end{bmatrix} \tag{1.6}$$

Here are three tasks:

- Compute the eigenvalues of A_2
- Pick your favorite two vectors in $u, v \in \mathbb{R}^2$
- Compute their L^2 norms between u and v
- Compute the quadratic form $Q(u, u)$ and $Q(v, v)$ for A_2

1.5 Theoretical Practice: Statement Proofs

Given square matrices A and B, it is a fact that $det(AB) = det(A)det(B)$. Use this fact to prove the following statements

$$det(A^{-1}) = \frac{1}{det(A)} \tag{1.7}$$

$$det(A^n) = (det(A))^n \tag{1.8}$$

Hint: use the fact that $det(I_n) = 1$, where I_n is the identity matrix of dimension n, for all positive integers n.

1.6 Numerical Experiment: Raising Matrix Powers

For a given matrix $A \in \mathbb{R}^{n \times n}$, suppose we can find n distinct eigenvectors, $v_1, ..., v_n$ and eigenvalues $\lambda_1, ..., \lambda_n$. Then we can write what is called "eigen-decomposition"

$$A = V \Lambda V^{-1} \tag{1.9}$$

where V is just a stack of the n eigenvectors, ie. $V = [v_1, ..., v_n]$ and Λ is a diagonal matrix $\lambda_1, ..., \lambda_n$ filling the diagonal entries.

1.6.1 Part I

Understand eigendecomposition by reading it up online. For instance, the Wikipedia page is quite comprehensive

1.6.2 Part II

Eigendecomposition is helpful in many ways. In this exercise, we shall use it to take powers of matrices.

Prove that

$$A^k = V\Lambda^k V^{-1} \tag{1.10}$$

Notice that since Λ is a diagonal matrix, $\Lambda^k = diag(\lambda_1^k, ..., \lambda_n^k)$. This is very helpful: instead of touching every entry of the $n-by-n$ matrix k times, we only need to touch n numbers (ie. the eigenvalues) one time by raising it to a power of k. Thus, it saves a lot of computational power.

1.6.3 Part III

Verify that (1.10) is true by doing the following numerical experiment.

- Generate a random matrix A
- Compute its eigenvalues and eigenvectors (In MATLAB and Python, eig(A) would do the job nicely
- Compute the left-hand side and right-hand side of (1.10) and compare the results

As a bonus, you can time the operations of the left side and right side of (1.10) and observe the respective computational cost for large n's.