Damian Franco
dfranco24@unm.edu
101789677
CS-575

Homework 3

1) For the first five questions we were asked to prove various norms stating whether they are indeed a norm or specific properties related to other vector or induced matrix norms. Each proof that asks us to prove if a norm is indeed a norm has three parts for the proof which you will see in the hand written proof below. Other proofs ask for specific norm properties to be proven. I used many properties and definitions of vector norms, matrix norms and induced matrix norms to complete these proofs. Below you can view my work for each proof.

1) Show $\|\cdot\|_\infty$ is indeed a norm

$$\|\cdot\|_\infty = \max_{1 \le i \le m} |x_i|$$

(a) Non-negativity/zero vector:

$\|x\| \geq 0$ and $\|x\| = 0$ only if $x = 0$

$\Rightarrow \|x\| = 0$

$\Rightarrow \max_{1 \le i \le m} |x_i| = 0$

$\Rightarrow \max_{1 \le i \le m}(|x_1|, |x_2|, \ldots, |x_m|) = 0$, where $x = 0$

$\Rightarrow \max_{1 \le i \le m}(|0|, |0|, \ldots, |0|) = 0$

$\Rightarrow \|0\|_\infty = 0$

therefore $\|0\|_\infty = 0$ ✓ DONE

(b) Homogeneity:

$$||cx||_\infty = |c| \cdot ||x||_\infty$$

$\Rightarrow ||cx||_\infty$

$\Rightarrow \max_{1 \le i \le m} |cx_i|$

$\Rightarrow |c| \cdot \max_{1 \le i \le m} |x_i|$

$\Rightarrow |c| \cdot ||x||_\infty$

therefore $||cx||_\infty = |c| \cdot ||x||_\infty$ $\checkmark$ DONE

(c) Triangle Inequality:

$$||x+y||_\infty = ||x||_\infty + ||y||_\infty$$

$\Rightarrow ||x+y||_\infty$

$\Rightarrow \max_{1 \le i \le m} |x_i + y_i|$

$\Rightarrow \max_{1 \le i \le m} (|x_i| + |y_i|)$

$\Rightarrow \max_{1 \le i \le m} |x_i| + \max_{1 \le i \le m} |y_i|$

$\Rightarrow ||x||_\infty + ||y||_\infty$

therefore $||x+y||_\infty = ||x||_\infty + ||y||_\infty$ $\checkmark$ DONE

2)

2) Show $||x||_\infty \le ||x||_2 \le \sqrt{m}\,||x||_\infty$

(i) Show $||x||_2 \le \sqrt{m}\,||x||_\infty$

$\Rightarrow ||x||_2$

$\Rightarrow \sqrt{x \cdot x}$

$\Rightarrow \sqrt{\sum_{i=1}^{m} |x_i|^2}$

$\Rightarrow \sqrt{\sum_{i=1}^{m} |x_i|^2 |1|}$

$\Rightarrow \sqrt{\sum_{i=1}^{m} |x_i|^2 \begin{bmatrix} |1| \\ \vdots \\ |1| \end{bmatrix}} \le \sqrt{\sum_{i=1}^{m} ||x_i||^2 \cdot \begin{bmatrix} |1| \\ \vdots \\ |1| \end{bmatrix}}$

$\le \sqrt{\sum_{i=1}^{m} ||x||_2^2 \cdot \sum_{i=1}^{m} 1^2}$

$\le \sqrt{\sum_{i=1}^{m} ||x_i||_\infty^2} \cdot \sqrt{\sum_{i=1}^{m} 1^2}$

$\le ||x||_\infty \cdot \sqrt{m}$

$\le \sqrt{m}\,||x||_\infty$

therefore $||x||_2 \le \sqrt{m}\,||x||_\infty$ ✓ DONE

(ii) Show $||x||_\infty \le ||x||_2$

$\Rightarrow ||x||_\infty$

$\Rightarrow \max_{1 \le i \le m} |x_i| \le \sqrt{\sum_{i=1}^{m} |x_i|^2}$

$\le ||x||_2$

therefore $||x||_\infty \le ||x||_2$ ✓ DONE

3)

3) Show that the induced matrix norm is a norm

$$||A||^{(m,n)} = \max_{x \in \mathbb{R}} \frac{||Ax||^{(m)}}{||x||^{(n)}}$$

(a) Non-negative/zero vector:

$||A||^{(m,n)} \geq 0$ if $A \neq 0$

$\Rightarrow ||A||^{(m,n)}$

$$\Rightarrow \max_{x \in \mathbb{R}} \frac{||Ax||^{(m)}}{||x||^{(n)}} \geq \frac{||A \sum_{i=1}^{m} |x_i|||}{||\sum_{i=1}^{m} |x_i|||}$$

$$\geq \frac{||Ax_i||}{||x_i||}$$

$$\geq 0$$

therefore $||A||^{(m,n)} \geq 0$ ✓DONE

(b) Homogeneity:

$||cA||^{(m,n)} = |c| \cdot ||A||^{(m,n)}$

$\Rightarrow ||cA||^{(m,n)}$

$$\Rightarrow \max_{x \in \mathbb{R}} \frac{||cAx||^{(m)}}{||x||^{(n)}}$$

$$\Rightarrow \max_{x \in \mathbb{R}} |c| \cdot \frac{||Ax||^{(m)}}{||x||^{(n)}}$$

$$\Rightarrow |c| \max_{x \in \mathbb{R}} \frac{||Ax||^{(m)}}{||x||^{(n)}}$$

$\Rightarrow |c| \cdot ||A||^{(m,n)}$

therefore $||cA)^{(m,n)} = |c| \cdot ||A||^{(m,n)}$ ✓DONE

(c) Triangle Inequality

$$||A+B||^{(m,n)} = ||A||^{(m,n)} \cdot ||B||^{(m,n)}$$

$$\Rightarrow ||A+B||^{(m,n)}$$

$$\Rightarrow \max_{x \in \mathbb{R}} \frac{||(A+B)x||^{(m)}}{||x||^{(n)}}$$

$$\Rightarrow \max_{x \in \mathbb{R}} \frac{||Ax+Bx||^{(m)}}{||x||^{(n)}} \leq$$

$$\Rightarrow \max_{x \in \mathbb{R}} \frac{||Ax||^{(m)} + ||Bx||^{(m)}}{||x||^{(n)}}$$

$$\Rightarrow \max_{x \in \mathbb{R}} \left(\frac{||Ax||^{(m)}}{||x||^{(n)}} + \frac{||Bx||^{(m)}}{||x||^{(n)}} \right)$$

$$\Rightarrow \max_{x \in \mathbb{R}} \frac{||Ax||^{(m)}}{||x||^{(n)}} + \max_{x \in \mathbb{R}} \frac{||Bx||^{(m)}}{||x||^{(n)}}$$

$$\Rightarrow ||A||^{(m,n)} + ||B||^{(m,n)}$$

therefore $||A+B||^{(m,n)} = ||A||^{(m,n)} + ||B||^{(m,n)}$

✓DONE

4)

4) Let $A \in \mathbb{R}^{m \times n}$

Show $||A||_\infty = \max_{1 \le i \le m} \sum_{j=1}^{n} |a_{ij}|$

$||A||_\infty = \max_{||x||_\infty = 1} ||Ax||_\infty$ or $\max_{x \in \mathbb{R}} \frac{||Ax||_\infty}{||x||_\infty}$

Let $||A||_\infty = 1$

$\Rightarrow ||A||_\infty$

$\Rightarrow \max_{||x||_\infty = 1} ||Ax||_\infty$

$$\Rightarrow \max_{||x||_\infty = 1} \left|\left| \begin{bmatrix} a_{11} \\ a_{22} \\ \vdots \\ a_{mm} \end{bmatrix} [x_1, x_2, \ldots x_j] \right|\right|$$

$$\Rightarrow \max_{||x||_\infty = 1} \left| \sum_{i=1}^{n} a_{ij} x_j \right| \le \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}| \cdot ||x_j||$$

$$\le \max_{1 \le i \le m} \sum_{j=1}^{n} |a_{ij}| \cdot \sum_{j=1}^{n} ||x_j||$$

$$\le \max_{1 \le i \le m} \sum_{j=1}^{n} |a_{ij}| \cdot 1$$

$$\le \max_{1 \le i \le m} \sum_{j=1}^{n} |a_{ij}|$$

therefore $||A||_\infty = \max_{1 \le i \le m} \sum_{j=1}^{n} |a_{ij}|$

✓DONE

5)

5) (i) Show $||I|| = 1$ where I is a $\mathbb{R}^{m \times m}$ identity matrix.

$\Rightarrow ||I||^{(m,m)}$

$\Rightarrow \max_{||x||=1} ||Ix||^{(m)}$

$\Rightarrow \max_{||x||=1} ||x||^{(m)}$

$\Rightarrow 1$

therefore $||I|| = 1$ ✓DONE

(ii) $||Ax||^{(m,m)} \leq ||A||^{(m,m)} \cdot ||x||^{(m)}$ for all $A \in \mathbb{R}^{m \times m}, x \in \mathbb{R}^m$

$\Rightarrow ||Ax||^{(m,m)}$

$\leq ||A(1/||x|| \cdot x)\, x||^{(m,m)}$

$\leq ||A(1/||x|| \cdot x)||^{(m,m)} \cdot ||x||^{(m)}$

$\leq ||Ax||^{(m,m)} \cdot ||x||^{(m)}$

therefore $||Ax|| \leq ||A||^{(m,m)} \cdot ||x||^{(m)}$ ✓DONE

(iii) $||AB|| \leq ||A|| \cdot ||B||$ for all $A, B \in \mathbb{R}^{m \times m}$

$\Rightarrow ||AB||$

$\Rightarrow \max_{||x|| \leq 1} ||ABx|| \leq \max_{||x|| \leq 1} ||A|| \cdot ||Bx||$

$\leq \max_{||x|| \leq 1} ||A|| \cdot \max_{||x|| \leq 1} ||Bx||$

$\leq ||A|| \cdot \max_{||x|| \leq 1} ||Bx||$

$\leq ||A|| \cdot ||B \cdot 1||$

$\leq ||A|| \cdot ||B||$

therefore $||AB|| \leq ||A|| \cdot ||B||$ ✓DONE

6) For the last question we were asked to create some code for solving a linear system using Gaussian Elimination without partial pivoting. This was similar to the previous homework question, but would instead be optimized to perform the algorithm on a banded matrix. The banded matrix is a tridiagonal matrix that has 10's on the diagonal, -4's on the super-diagonal and -2's on the sub-diagonal. We were required to make a N x 3 matrix that contains the values in our tridiagonal matrix and use that in a function to create a dense version of the matrix that contains zeros in all other places that are not on the tridiagonal positions. I utilized Professor Zeb's example code for creating a banded matrix for this part of the question. Next, we were asked to create a vector for the right-hand side (b) of the system that was all 1's and of size N x 1. After this was done, I took my code from the previous homework that performs forward elimination and backward substitution and optimized it to only parse through the indices that contained non-zero values. I then used the updated versions of the GE without substitution method and estimated the solution to the linear system with the size of N = 4 and various N sizes to test out performance and accuracy. I found that this version of GE without partial pivoting is very fast compared to the non-optimized version when compared against the optimized version that I created. That is expected because parsing through zero values is not ideal performance-wise. I also had unexpected 0 accuracy on all my tests of matrices and vectors of even large N sizes. This seems somewhat like an impossibility but somehow through various tests, I was not able to break the accuracy on it. I looked into this and found that banded matrices could possibly improve performance, but I still find it hard to believe that this is occurring without some sort of concrete explanation. I will definitely look into this more. Overall, I noticed that optimizing code can ultimately lead to overall improvements to performance and in my case, even accuracy. On the next couple of pages, you will be able to view my work alongside some statistics and a plot of my performance against the bound O(n) which shows some parallelism with its bound and the measured runtime.

Function 'create_tri_mat' from Professor Zeb's example:

```
def create_tri_mat(N):
  A_tri= np.zeros((N,3))
  A_tri[1:N,0] = -2*np.ones((N-1,)) #sub-diagonal
  A_tri[0:N,1] =  10*np.ones((N,))   #diagonal
  A_tri[0:N-1,2] = -4*np.ones((N-1,))   #super-diagonal
  return A_tri
```

Function 'create_dense_mat' from Professor Zeb's example:

```
def create_dense_mat(A_tri,N):
  A_dense = np.zeros((N,N))
  for i in range(N): # go over all rows
    # only 3 non-zero entries in each row.
    for j in range(i-1,i+2):  #This will go through entries (i,i-1), (i,i) and (i,i+1)
      if j >= 0 and j <= N-1:
        A_dense[i,j] = A_tri[i,j-i+1] #Make sure you understand the indexing
  return A_dense
```

Function 'forward_tri':

```
def forward_tri(A, b):
    n = len(b)
    for k in range(n-1):
        xmult = A[k+1][k] / A[k][k]
        A[k+1][k+1] = A[k+1][k+1] - xmult * A[k][k+1]
        b[k+1] = b[k+1] - xmult * b[k]
    for k in range(n):
        for i in range(max(0, k-1), min(n, k+2)):
            if i != k+1:
                continue
            xmult = A[i][k] / A[k][k]
            for j in range(max(0, k-1), min(n, k+2)):
                if j != k and j != i:
                    continue
                A[i][j] = A[i][j] - xmult * A[k][j]
            b[i] = b[i] - xmult * b[k]
    return A, b
```

Testing with N = 4 Matrix:

```
N = 4
A_tri = create_tri_mat(N)
print(A_tri)
```

```
[[ 0. 10. -4.]
 [-2. 10. -4.]
 [-2. 10. -4.]
 [-2. 10.  0.]]
```

```
A_dense = create_dense_mat(A_tri, N)
print(A_dense)
```

```
[[10. -4.  0.  0.]
 [-2. 10. -4.  0.]
 [ 0. -2. 10. -4.]
 [ 0.  0. -2. 10.]]
```

```
b = np.ones(N)
print(b)
```

```
[1. 1. 1. 1.]
```

```
aprroxGE = GE(A_dense, b)
print("My approximation:", aprroxGE)
print("Exact solution:", exact)
```

```
My approximation: [0.1868476  0.217119   0.1993737  0.13987474]
Exact solution: [0.1868476  0.217119   0.1993737  0.13987474]
```

Testing with various N sizes:

```
N_arr = [100,200,300,400,500,600,700,800,1000]
```

N size	Runtime Speed (sec)	Error
100	0.005667686462402344	0.0
200	0.021142244338989258	0.0
300	0.051323652267456055	0.0
400	0.10359001159667969	0.0
500	0.12713623046875	0.0
600	0.15452265739440918	0.0
700	0.20960474014282227	0.0
800	0.22070765495300293	0.0
1000	0.29892659187316895	0.0