Juan Franco EECE 212

Lab 6 Linear Algebra Functions in MATLAB

Lab6 %Running Lab 6 file

A.\*B %Multiplying A and B

Ans =

8 2 -2 %Answer for multiplying A and B

c %Multiplying A and transpose of B

c = %Values for c

8 -2 4

-8 2 -4

-4 1 -2

C %Multiplying transpose of A and A

C = %Values for C

16 -4 8

-4 1 -2

8 -2 4

r = rref(C) %Checking rank of C

r = %Value for rank of C

1 -0.2500 0.5000

0 0 0

0 0 0

Lab6 %Running Lab 6 file

C\_noise %C adding 0.0001 to C(1,2) and C(2,3)

C\_noise = %Values of C\_noise

16.0000 -4.0001 8.0000

-4.0000 1.0000 -2.0001

8.0000 -2.0000 4.0000

r = rref(C\_noise) %Checking rank of C\_noise

r = %Values for rank of C\_noise

1 0 0

0 1 0

0 0 1

rank(C) %Checking rank of C

Ans = %Value for rank of C

1

rank(C\_noise) %Checking rank of C\_noise

Ans = %Value for rank of C\_noise

3

**% Rank(C) is 1 because it is linearly dependent while Rank(C\_noise) is 3 as it % is linearly independent. I would use Rank(C) as since it is only rank 1 it would % be easier.**

Lab6 %Running Lab 6 file for my\_elimination function

16 -4 8 %Values for U1

0 0 0

0 0 0

8.0000 1.0000 6.0000 %Values for U2

0 4.6250 4.7500

0 0 -9.7297

1 0 0 %Values for U3

0 1 0

0 0 1

C^(-1) %Checking inverse of C

[�Warning: Matrix is singular to working precision.]� %Warning appears as we are

%dividing a value by 0 which gives Inf

Ans = %Values for inverse of C

Inf Inf Inf

Inf Inf Inf

Inf Inf Inf

C\_noise^(-1) %Checking inverse of C\_noise

Ans = %Values for inverse of C\_noise

1.0e+04 \*

-0.2500 0.5000 0.7500

-1.0000 0 2.0000

0 -1.0000 -0.5000

[U,S,V] = svd(C\_noise) %Checking for matrices U,S and V from C\_noise

U = %Values for matrix U

-0.8729 -0.2731 -0.4044

0.2182 -0.9597 0.1770

-0.4364 0.0663 0.8973

S = %Values for matrix S

21.0000 0 0

0 0.0001 0

0 0 0.0000

V = %Values for matrix V

-0.8729 -0.3369 0.3530

0.2182 0.3776 0.8999

-0.4364 0.8625 -0.2560

[L,U,P] = lu(C) %Checking for matrices L,U and P from C

L = %Values for matrix L

1.0000 0 0

-0.2500 1.0000 0

0.5000 0 1.0000

U = %Values for matrix U

16 -4 8

0 0 0

0 0 0

P = %Values for matrix P

1 0 0

0 1 0

0 0 1

N = null(C) %Checking null space for C

N = %Value for null space

-0.4880 0

-0.3904 -0.8944

0.7807 -0.4472

C \* N %Multiplying C and N

Ans = %Values of C times N

1.0e-14 \*

0.0888 0.1332

-0.0222 -0.0333

0.0444 0.0666

Explore\_An %Running Explore\_An file

diary off %Turning diary off

Graph of **“D to the power 1~100”**



Graph of **“E to the power 1~100”**



Graph of **“F to the power 1~100”**



**I noticed that for the graph of the D matrix, the rows added up both equal to one. For the graph of E the top row adds up to 0.9 and the graph of F the top row adds up to 1.1. Based off that I can say that if the rows add up to 1, then they will converge at two different y-line values. If they are less than 1 then they will all converge going to 0. If they are more than 1 than it will not converge and continue to increase.**