COL 726 Assignment 2

K LAXMAN

TOTAL POINTS

50.5 / 75

QUESTION 1

1 1 **10 / 10**

- **- 10 pts** InCorrect

✓ **- 0 pts** *Correct*

- **- 2 pts** Not Said about SVD and Normalization
- **- 2 pts** Not said/Explained about rank and its relation with answer
- **- 2 pts** Not said/ Explained about sequence of matrices
- **- 3 pts** Not said about $$\left || B^{k}-A \right || -> 0$$ as {k->inf}
- **- 1.5 pts** Not Given Mathematical Proof
- **- 2 pts** Proper Explanation not Given- Missing Reasons in most of the parts

QUESTION 2

2 2 **0 / 15**

- **- 15 pts** Definition of Backward Stability is itself incorrect.
- **- 0 pts** Correct Proof of Backward Stability of the Algorithm: Shows that $$\exists x', y'$$, such that $$<x', y'> = S, x'$$ ~ $$x, y'$$ ~ $$y$$
- **- 0 pts** Correct Proof of Backward Stability of the Matrix-Vector Multiplication Algorithm
- **- 2 pts** Does not show that $$||\tilde{A} - A||/||A|| = O(\epsilon)$$
- **- 5 pts** Incomplete/Incorrect Proof of Backward Stability of the Algorithm: Does not show that $$\exists x', y'$$, such that $$<x', y'> = S, x'$$ ~ $$x, y'$$ ~ $$y$$

✓ **- 15 pts** *No Attempt*

- **- 4 pts** Incomplete/Incorrect Proof of Backward Stability of the Matrix-Vector Multiplication Algorithm
- **- 8 pts** Incomplete/Incorrect Proof of Backward Stability of the Algorithm: Does not show that $$\exists x', y'$$, such that $$<x', y'> = S, x'$$ ~ $$x, y'$$ ~ $$y$$

QUESTION 3

3 3 **4.5 / 10**

- **- 0 pts** Correct
- **- 10 pts** Incorrect

✓ **- 7 pts** *Not Given Mathematical Proof*

- **- 3 pts** Not Explained Correctly / Not mentioned dimensions / Taken Assumptions / Solved by example
- **- 3 pts** Not written bout ||Cv|| -> ||ABv||
- **- 3 pts** Not mentioned ||Ay||>=$$a_{n}$$||y||
- **- 3 pts** Not explained - how we get bi

+ 1.5 *Point adjustment*

1

QUESTION 4

4 4 **7 / 10**

- **- 0 pts** Correct

- 10 pts Incorrect

- 1.5 pts Not mentioned about singular Values SVD / Proper reason not given for each step / Not shown relation between ||A||$$_{F}$$ and ||A^{T}||$$_{F}$$

- 3 pts Not shown summation equation and its solving

- 3 pts Proper Reasoning Needed - Step by step

✓ **- 3 pts** *Not shown derivation of $$||B||_{F}$$*

QUESTION 5

5 5 **0 / 0**

✓ **- 0 pts** *Ok*

QUESTION 6

6 6 **10 / 10**

✓ **- 0 pts** *Correct*

- 1 pts Missing Details

- 3 pts Missing Various Details

- 10 pts Incorrect/No Attempt

QUESTION 7

7 7 **5 / 5**

✓ **- 0 pts** *Correct*

- 3 pts Incomplete/Incorrect Code for `house(A)`

- 2 pts Incomplete/Incorrect Code for `formQ(W)`

- 5 pts No Attempt

QUESTION 8

8 8 **5 / 5**

✓ **- 0 pts** *Correct Code for `leastSquare` Method*

- 5 pts Incorrect/Incomplete Code for `leastSquare` Method

- 5 pts No Attempt

QUESTION 9

9 9 **9 / 10**

- 0 pts Correct

- 5 pts (a) Incorrect Values/Plot

- 5 pts (a) No Plot/Values

- 3 pts (a) No Plot

- 5 pts (b) Incorrect Plot

✓ **- 1 pts** *(b) Does Not Show that the Problem is ill-conditioned, leading to difference in Plots.*

- 5 pts (b) No Plot

- 10 pts No Attempt

Assignment - 2

COL 726

(1) The proof that any matrix can be approximated by a sequence of matrices of full rank can be done using SVD

consider an arbitrary matrix $A_{m\times n}$ and its SVD is $A = U\Sigma V^*$

where U having order $m\times m$ and V having order $n\times n$ are unitary matrices

Let us assume that m is greater then equal to n i.e. $m \geq n$

The singular values of matrix A are denoted by

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \dots \sigma_n \geq 0$$

now we need to construct a sequence of matrices

$$\{A_k\}_{k=1}^{\infty} \text{ as}$$

$$A_k = U\Sigma_k V^*$$

where

$$\Sigma_k = \begin{bmatrix} \sigma_1 + 1/k & & & \\ & \sigma_2 + 1/k & & \\ & & \ddots & \\ & & & \sigma_n + 1/k \end{bmatrix} \text{ having order } (m\times n)$$

now, $\forall\ k \in \mathbb{N}$, we have

$$\text{rank}(A_k) = \text{rank}(U\Sigma_k V^*) = \text{rank}(\Sigma_k) = n$$

So, and hence we can say $\{A_k\}_{k=1}^{\infty}$ are a set of full rank matrices.

Now using the 2-norm of the matrix A

$$\|A - A_k\|_2 = \|U(\Sigma - \Sigma_k)V^*\|_2$$
$$= \|\Sigma - \Sigma_k\|_2$$
$$= \sqrt{\rho[(\Sigma - \Sigma_k)^*(\Sigma - \Sigma_k)]}$$
$$= 1/k$$

Since here, Σ is a diagonal matrix, and $\|A - A_k\|_2 \to 0$ (approaches zero as k approaches 0). Which implies that set of full rank matrices is a dense subset of $C^{m \times n}$. it shows matrix can be approximated arbitrarily closely to sequence of full rank.

1 1 **10 / 10**

- **- 10 pts** InCorrect
- ✓ **- 0 pts** *Correct*
- **- 2 pts** Not Said about SVD and Normalization
- **- 2 pts** Not said/Explained about rank and its relation with answer
- **- 2 pts** Not said/ Explained about sequence of matrices
- **- 3 pts** Not said about $$\left || B^{k}-A \right || -> 0$$ as {k->inf}
- **- 1.5 pts** Not Given Mathematical Proof
- **- 2 pts** Proper Explanation not Given- Missing Reasons in most of the parts

2 2 **0 / 15**

- 15 pts Definition of Backward Stability is itself incorrect.

- 0 pts Correct Proof of Backward Stability of the Algorithm: Shows that $$\exists x', y'$$, such that $$<x', y'> = S, x'$$ ~ $$x, y'$$ ~ $$y$$

- 0 pts Correct Proof of Backward Stability of the Matrix-Vector Multiplication Algorithm

- 2 pts Does not show that $$||\tilde{A} - A||/ ||A|| = O(\epsilon)$$

- 5 pts Incomplete/Incorrect Proof of Backward Stability of the Algorithm: Does not show that $$\exists x', y'$$, such that $$<x', y'> = S, x'$$ ~ $$x, y'$$ ~ $$y$$

✓ **- 15 pts** *No Attempt*

- 4 pts Incomplete/Incorrect Proof of Backward Stability of the Matrix-Vector Multiplication Algorithm

- 8 pts Incomplete/Incorrect Proof of Backward Stability of the Algorithm: Does not show that $$\exists x', y'$$, such that $$<x', y'> = S, x'$$ ~ $$x, y'$$ ~ $$y$$

Q③:- To prove

$$c_i \geq b_i . a_n$$

As given that A and B be two square $n \times n$ matrices

Let $C = AB$. Let $c_1 \geq c_2 \geq \cdots c_n$

From the given above, we know that c_i's are square root of eigen values of C^TC.

$C^TC = (AB)^T(AB) = B^TA^TAB$

$\lambda_i(C^TC) = \lambda_i(B^TA^TAB)$

As we know that eigen value of product of matrices = eigen value of product of their transposes

ie. $\lambda_i(C^TC) = \lambda_i(B^TA^TAB)$
$\lambda_i(A^TAB^TB)$

Now, consider that c_i and singular vectors of corresponding are u_i, v_i

$c_i^2 = \|Cv_i\|^2$
$= \|ABv_i\|^2$ here v_i is the right singular & u_i is left

From above
$\|ABv_i\|^2 = \|c_iU_i\|^2 = c_i^2\|U_i\|^2$

by considering the ith row of B and nth column of A & by using Cauchy schwarz inequality

$|b_i^Ta_n| = |(a_n^Tb_i)^T| \leq \frac{\|a_n\|}{c_i} \cdot \|b_i\|$

$\therefore c_i^2\|U_i\|^2 \geq (b_i a_n)^2$

$c_i \geq \frac{b_i a_n}{\|U_i\|}$

Hence $u_i = 1 \Rightarrow c_i \geq b_i a_n$

Hence proved

3 3 **4.5 / 10**

- 0 pts Correct

- 10 pts Incorrect

✓ **- 7 pts** *Not Given Mathematical Proof*

- 3 pts Not Explained Correctly / Not mentioned dimensions / Taken Assumptions / Solved by example

- 3 pts Not written bout ||Cv|| -> ||ABv||

- 3 pts Not mentioned ||Ay||>=$$a_{n}$$||y||

- 3 pts Not explained - how we get bi

+ 1.5 *Point adjustment*

1

Qs ④ :-

K. Laxman
2018CS50408

To prove :- $\|ABC\|_F \leq \|A\|_2 \|B\|_F \|C\|_2$ for the A, B, C matrices

Initially to prove the main claim, firstly I will prove for any 2 matrices A, B of appropriate dimensions

* For any 2 matrices A, B of appropriate dimensions $\|AB\|_F \leq \|A\|_2 \|B\|_F$ and also $\|AB\|_F \leq \|A\|_F \|B\|_2$

by using SVD i.e. $A = U\Sigma V^T$.

$AB = U\Sigma V^T B$

The Frobenius form of 'AB' $= \|U\Sigma V^T B\|_F = \|\Sigma V^T B\|_F$.

suppose $V^T B = C$. and c_i be the diagonal element of 'C' & 'σ_i' be singular value of Σ. i.e. $\sigma_1 > \sigma_2 > \dots \sigma_n$

$\therefore \|\Sigma C\|_F = \sqrt{\sum_i \sigma_i c_i^2}$

$\Rightarrow \|\Sigma C\|_F \leq \sqrt{\sigma_1^2 \sum c_i^2}$

i.e. $\|\Sigma\|_2 \|C\|_F$ Note σ_1 is the greatest

So, $\|\Sigma V^T B\| \leq \|\Sigma\|_2 \|V^T B\|_F = \|\Sigma\|_2 \|B\|_F$

In the similar case we can do for, already we know that $\|A\|_F = \|A^T\|_F$.

Same apply for $\|AB\|_F = \|(AB)^T\|_F = \|B^T A^T\|_F \rightarrow ①$

From ① by simplification

$\Rightarrow \|AB\|_F \leq \|A\|_F \cdot \|B\|_2$

So, as above said for supporting we have proved that.

From above claim. we have

$\|ABC\|_F \leq \|A\|_2 \|BC\|_F$,

by using the same claim from above similarly,

$\|ABC\|_F \leq \|A\|_2 \|B\|_F \|C\|_2$ for A, B, C matrices.

4 4 **7 / 10**

- **- 0 pts** Correct
- **- 10 pts** Incorrect
- **- 1.5 pts** Not mentioned about singular Values SVD / Proper reason not given for each step / Not shown relation between ||A||$$_{F}$$ and ||A^{T}||$$_{F}$$
- **- 3 pts** Not shown summation equation and its solving
- **- 3 pts** Proper Reasoning Needed - Step by step
- ✓ **- 3 pts** *Not shown derivation of $$||B||_{F}$$*

K.Laxman 2018CS50408

⑤ To get the QR factorisation by using CGS method to solve the linear system of equation of expression $Ax = b$

$\vec{v}_i = \vec{q}_i \quad \forall i$

for $i = 1 \ldots n$:

for $k = 1 \ldots i$:

$v_i = \vec{q}_k \underbrace{\vec{q}_k \vec{a}_i}_{r_{ki}}$

$$\vec{q}_i = \frac{\vec{v}_i}{\|v_i\|} \leftarrow r_{ii}$$

Similarly for MGS

$\vec{v}_i = \vec{a}_i \;\forall i$

for $i = 1 \ldots n$

$r_{ii} = \|\vec{v}_i\|$

$$q_i = \frac{\vec{v}_i}{r_{ii}}$$

for $j = i+1 \ldots n$:

$r_{ij} = \vec{q}_i * \vec{v}_j$

$v_j -= r_{ij} \vec{q}_i$

given the matrix

$$A = \begin{bmatrix} 1 & 1 & 1 \\ \mu & 0 & 0 \\ 0 & \mu & 0 \\ 0 & 0 & \mu \end{bmatrix} \quad \vec{a}_1 = \begin{bmatrix} 1 \\ \mu \\ \\ \end{bmatrix} \quad \vec{a}_2 = \begin{bmatrix} 1 \\ \\ \mu \\ \end{bmatrix} \quad \vec{a}_3 = \begin{bmatrix} 1 \\ \\ \\ \mu \end{bmatrix}$$

$\mu = O(\sqrt{\varepsilon_m}$

In further computation, I have ~~taken~~ ignored terms $\leq O(\varepsilon_m^2)$

a) $r_{11} = \|\vec{O}_1\|_2 = \|\vec{a}_1\| = \|1 \cdot \vec{e}_1 + \mu \vec{e}_2\|$

$\tilde{r}_{11} = |(1 \oplus \vec{\mu} \odot \vec{\mu}) \oslash \frac{1}{2}|$

$$= \left[\left(\left(1+\mu^2(1+\varepsilon_1)\right)^2(1+\varepsilon_2)\right)(1+\varepsilon_3)\right]^{1/2}(1+\varepsilon_4)\Big)$$

here $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4 \le \varepsilon_m$

$$\tilde{r}_{11} = \left|\left((1+\mu^2)(1+\varepsilon_3)^{1/2}\right)(1+\varepsilon_4)\right|$$

$$= \left|\left(1+(\mu^2+\varepsilon_3)\right)^{1/2}\right|(1+\varepsilon_4)$$

$$= \left|\left(1+\tfrac{1}{2}(\mu^2+\varepsilon_3) - \tfrac{1}{3}(\mu^2+\varepsilon_3)^2\right|(1+\varepsilon_4)$$

$$= 1+\tfrac{1}{2}(\mu^2+\varepsilon_3)+\varepsilon_4$$

$$= 1+O(\varepsilon_m)$$

$$\tilde{q}_1 = \frac{\vec{v}_1}{r_{11}} = \frac{\vec{a}_1}{r_{11}} = \frac{1}{r_{11}}\begin{bmatrix}1\\ \mu\\ 0\\ 0\end{bmatrix}$$

$$\tilde{q}_1 = \begin{bmatrix} 1 \oslash \tilde{r}_{11}\\ \tilde{\mu} \oslash \tilde{r}_{11}\\ 0\\ 0\end{bmatrix} = \begin{bmatrix} 1/(1+O(\varepsilon_m))\times(1+\varepsilon_1)\\ \mu(1+\varepsilon_2)/(1+O(\varepsilon_m))\times(1+\varepsilon_2)\\ 0\\ 0\end{bmatrix}$$

$$\tilde{q}_1 = \begin{bmatrix} 1+O(\varepsilon_m)\\ \mu(1+O(\varepsilon_m))\\ 0\\ 0\end{bmatrix}$$

$$r_{12} = \tilde{q}_1 \cdot a_1$$

$$\tilde{r}_{12} \quad \tilde{q}_1 \circledast \tilde{q}_1$$

$$\tilde{q}_{11} \odot \tilde{a}_{11} \oplus \tilde{q}_{21} \odot q_{21}$$ here Note: $\tilde{a}_{11} = a_{11}$

$$= 1+O(\varepsilon_m)(1) \oplus \mu(1+O(\varepsilon_m))(\mu(1+\varepsilon_1))(1+\varepsilon_2).$$

by simplification

$$= 1+O(\varepsilon_m)$$

$$\tilde{q}_2 = \frac{\tilde{v}_2}{r_{22}}$$

$$r_{22} = \|\tilde{v}_2\| = \|a_2 - r_{12}\tilde{q}_2\|$$

$$\tilde{r}_{22} = \left\| \left(\begin{bmatrix} 1 \\ 0 \\ \mu \\ 0 \end{bmatrix} \ominus \tilde{r}_{12} \odot \begin{bmatrix} 1 + O(\varepsilon_m) \\ \mu + O(\mu\varepsilon_m) \\ 0 \\ 0 \end{bmatrix} \right) \right\|^{1/2} \longrightarrow ①$$

by simplifying $\tilde{r}_{22}$ i.e ①

$$= \left\| \begin{bmatrix} O(\varepsilon_m) \\ O(\mu\varepsilon_m) \\ O(\mu\varepsilon_m) \\ 0 \end{bmatrix} \right\|$$

$$= (O(\varepsilon_m^2))^2 (1+\varepsilon_m)$$

$$= O(\varepsilon_m)$$

$$\tilde{q}_2 = \frac{1}{r_{22}} \cdot \bar{v}_2$$

$$\tilde{q}_2 = O(\varepsilon_m^{-1})(1+\varepsilon_1) \odot \begin{bmatrix} O(\varepsilon_m) \\ O(\mu\varepsilon_m) \\ O(\mu\varepsilon_m) \\ 0 \end{bmatrix}$$

we can conclude that based on computed columns of Q in both cases, we can ~~conclu~~ say that classical gram-schmidt method produces a computed Q that is not very orthogonal while modified gram-schmidt produced computed Q ~~is~~ that is more orthogonal.

5 5 **0 / 0**

✓ **- 0 pts** *Ok*

⑥ Consider the matrix $\begin{bmatrix} I & F \\ 0 & I \end{bmatrix}$. here I is $n\times n$ matrix, F is $n\times n$ matrix.

To prove :- 2 norm of this matrix is equal to $\sqrt{2+\|F\|_2^2+\|F\|_2\sqrt{4+\|F\|_2^2}}$

proof :- The 2norm of this new vector will give information about matrices streches or compress vectors.

The 2 norm formule of matrix A = $\|A\|_2 = \max \frac{\|A.x\|_2}{\|x\|_2}$

Let us suppose the same matrix A and suppose $x = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$

here we can see that $\|x\|_2 = 1$ $(\sin^2\theta+\cos^2\theta)$. and suppose $B = \|F\|_2$

$$Ax = \begin{bmatrix} \cos\theta + B\sin\theta \\ \sin\theta \end{bmatrix}$$

2 norm of matrix A is $\|A\|_2 = \sqrt{(\cos\theta+B\sin\theta)^2+\sin\theta^2}$

$$= 1+K^2\sin^2\theta+2K\cos\theta\sin\theta \longrightarrow ①$$

now we have to find the value of 'θ' that maximises the expression which will give maximum strech of vector, this can be done by taking derivation of expression w.r.t. θ and set it to 0

by some simplifications to get maximise

from ① above

$$1+k^2\sin^2\theta+2k\cos\theta\sin\theta \qquad \because a\cos\theta+b\sin\theta = \sqrt{a^2+b^2}$$

$$1+k^2\left(\frac{1-\cos2\theta}{2}\right)+k\sin2\theta$$

$$1+\frac{k^2}{2}+\underbrace{k\sin2\theta + \frac{k^2}{2}\cos2\theta}_{\text{maximise of this}}$$

$$\Rightarrow 1+\frac{k^2}{2}+\sqrt{K^2+\left(\frac{K^2}{2}\right)^2} \qquad \because \text{here } B = K$$

$$= 1+\frac{k^2}{2}+\frac{k\sqrt{4+k^2}}{2}$$

$$\therefore = \sqrt{\frac{2+k^2+k\sqrt{4+k^2}}{2}}$$

hence

$$\|A\|_2 = \sqrt{\frac{2 + \|F\|_2^2 + \|F\|_2 \sqrt{\|F\|_2^2 + 4}}{2}}$$

hence proved

6 6 **10 / 10**

✓ **- 0 pts** *Correct*

- 1 pts Missing Details

- 3 pts Missing Various Details

- 10 pts Incorrect/No Attempt

K LAXMAN 2018CS50408

House.m

```
function [W,R] = house(A)
   R = A;
   [m,n] = size(R);
   W = zeros(m,n);
   for i = 1:n
       w = R(i:m,i);
       w1 = zeros(size(w));
       w1(1) = norm(w);
       w = w + w1.*sign(w(1));
       norm_val = norm(w);
       w = w/norm_val;
       x = w'*R(i:m,i:n);
       R(i:m,i:n) = R(i:m,i:n) - 2.*x.*w;
       W(i:m,i) = w;
   end
end
```

FormQ.m

```
function Q = formQ(W)
   [m,n] = size(W);
   Q = eye(m);
   for i = 1:n
       Q = Q*(eye(m)-2*(W(:,i)*W(:,i)'));
   end
end
```

leastSquare.m

```
function x = leastSquare(A,b)
[m,n] = size(A);
[W,R] = house(A);
Q = formQ(W);
R = R(1:n,:);
y = Q'*b';
y = y(1:n,:);
x = R\y;
end
```

Code :
https://drive.google.com/drive/folders/1Rt8mhS7rq4PC8Xj0j4bq7H1aV8QqaoEE?usp=sharing

7 7 **5 / 5**

- ✓ **- 0 pts** *Correct*
- **- 3 pts** Incomplete/Incorrect Code for `house(A)`
- **- 2 pts** Incomplete/Incorrect Code for `formQ(W)`
- **- 5 pts** No Attempt

Assign2.m

```
%% q1
clc;
clear;
close all;
A = [1 1;1 1.18;1 2.2];
[W,R] = House(A);
Q = formQ(W);
Q
R
%% q2
clc;
clear;
close all;
A = [1 1;1 1.18;1 2.2];
b = [1,2,3];
x = leastSquare(A,b)
%% q3
clc;
clear;
close all;
x1 = [1.02, 0.95, 0.87, 0.77, 0.67, 0.56, 0.44, 0.30, 0.16, 0.01];
y1 = [0.39, 0.32, 0.27, 0.22, 0.18, 0.15, 0.13, 0.12, 0.13, 0.15];
A = [y1'.^2, x1'.*y1', x1', y1', ones(size(x1'))];
b = x1.^2;
coeff = leastSquare(A,b);

x = linspace(-2, 2, 400);
y = linspace(-2, 2, 400);
[x,y] = meshgrid(x, y);

figure(1);
clf;
hold on;

contour(x, y, (coeff(1)*y.^2 + coeff(2)*x.*y + coeff(3)*x + coeff(4)*y + coeff(5) - x.^2));
disp(coeff(1)) % values of a ,b,c,d,e,
disp(coeff(2))
disp(coeff(3))
disp(coeff(4))
disp(coeff(5))
```

8 8 **5 / 5**

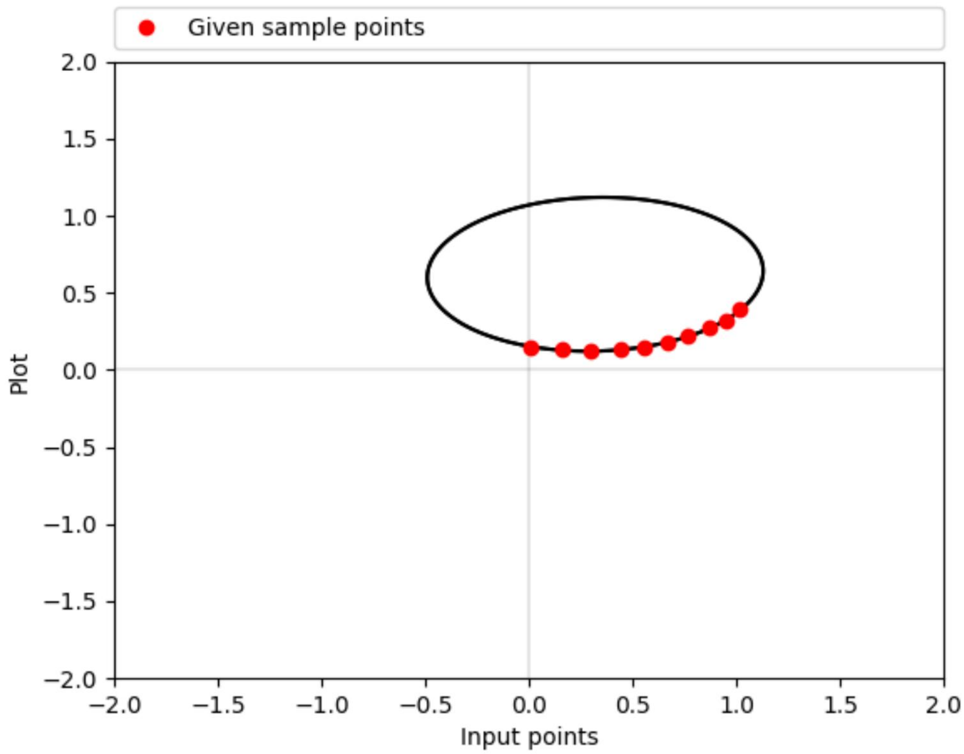
✓ **- 0 pts** *Correct Code for `leastSquare` Method*

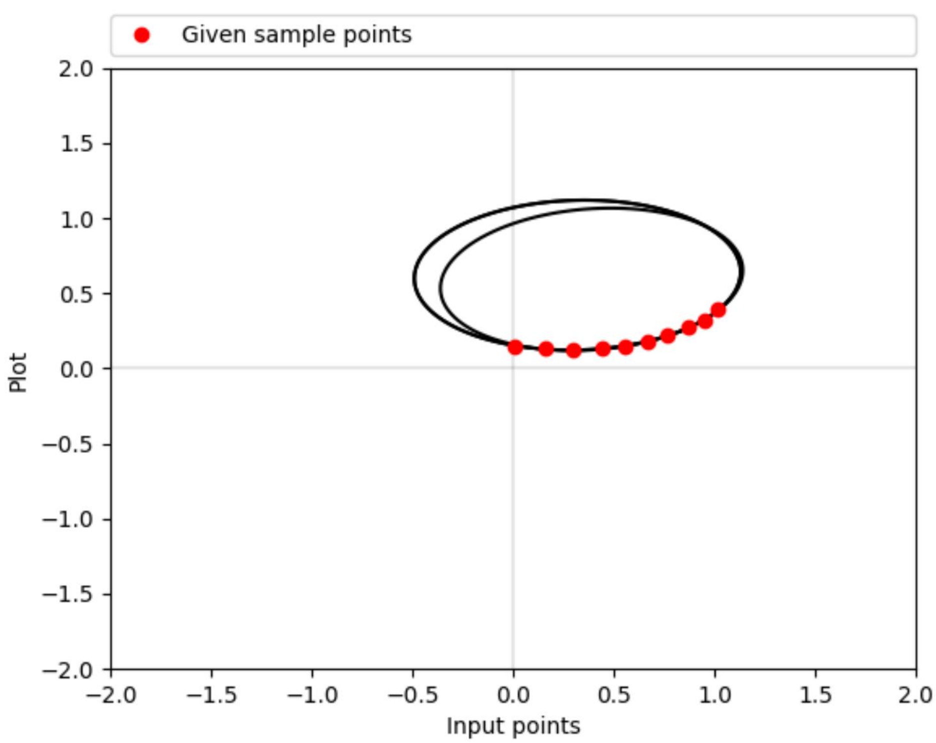
- 5 pts Incorrect/Incomplete Code for `leastSquare` Method

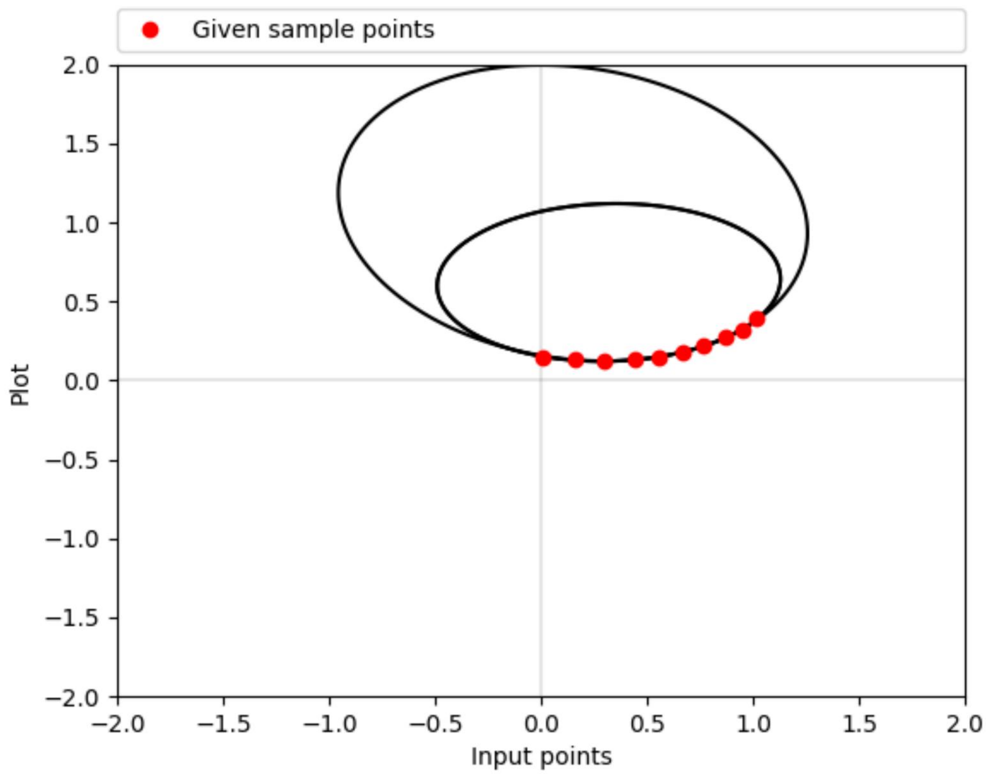
- 5 pts No Attempt

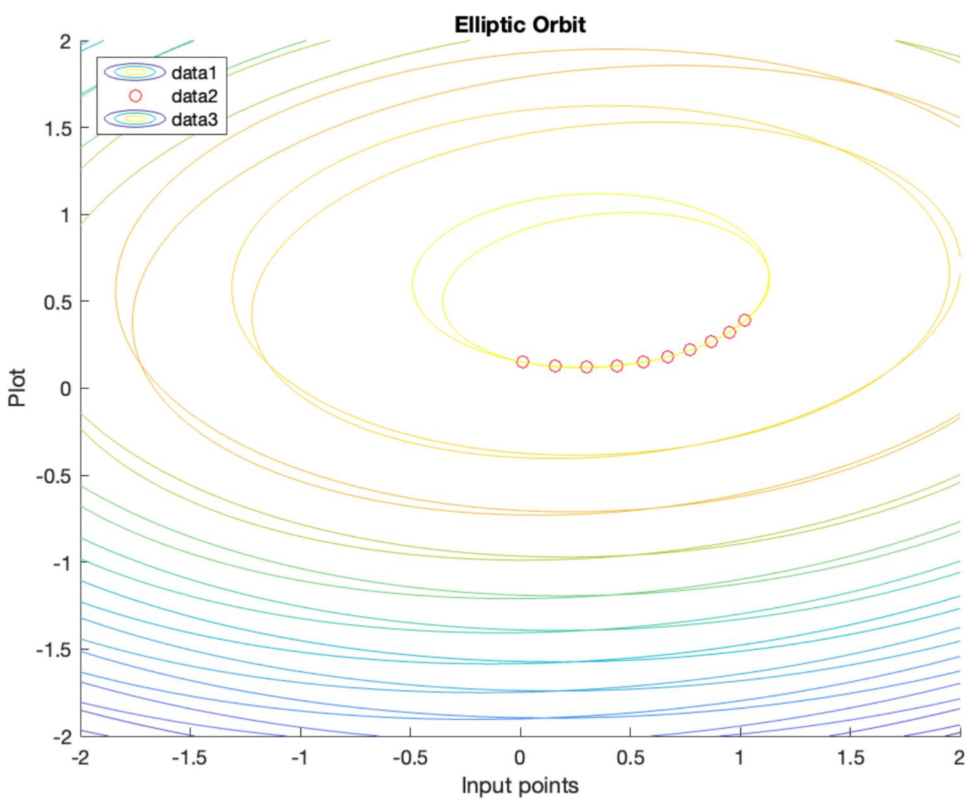
```
plot(x1, y1, 'ro');



% Set plot title and axis labels
title('Elliptic Orbit');
xlabel('Input points');
ylabel('Plot');

% Add legend
legend('show', 'Location', 'NorthWest');


% Part (b)
perturbx = rand(1,10)*0.01-0.005;
perturby = rand(1,10)*0.01-0.005;
x2 = x1+perturbx;
y2 = y1+perturby;

A1 = [y2'.^2, x2'.*y2', x2', y2', ones(size(x2'))];
b1 = x2.^2;

coeff1 = leastSquare(A1,b1);
contour(x, y, (coeff1(1)*y.^2 + coeff1(2)*x.*y + coeff1(3)*x + coeff1(4)*y + coeff1(5) - x.^2));
disp(coeff1(1)) % values of a ,b,c,d,e,
disp(coeff1(2))
disp(coeff1(3))
disp(coeff1(4))
disp(coeff1(5))
legend('show', 'Location', 'NorthWest');
hold off
disp(coeff);
disp(coeff1);
```

Individual Eclipse :






Given sample points
Plot
Input points
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


Elliptic Orbit
data1
data2
data3
Plot
Input points
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

q9) Explanation: When we perturb the input data slightly, the resulting least squares problem becomes well-conditioned. As a result, the new values of the parameters will be slightly different from the ones obtained previously.

The effect of this difference on the plot of the orbit will depend on the magnitude of the perturbations and the specific values of the parameters. In general, small perturbations should not affect the plot of the orbit significantly, and the orbit should still be a good fit to the data.

However, if the perturbations are large enough, they may cause the orbit to deviate significantly from the original data points. This can happen because small changes in the parameters can lead to large changes in the shape of the ellipse, especially if the ellipse is elongated or has a large eccentricity.

In some cases, the perturbations may even cause the ellipse to become unstable, meaning that the planet's orbit may not be predictable over long periods of time. This can happen if the perturbations push the orbit close to a resonance or other critical point in the planet's motion.

9 9 **9 / 10**

- **- 0 pts** Correct
- **- 5 pts** (a) Incorrect Values/Plot
- **- 5 pts** (a) No Plot/Values
- **- 3 pts** (a) No Plot
- **- 5 pts** (b) Incorrect Plot
- ✓ **- 1 pts** *(b) Does Not Show that the Problem is ill-conditioned, leading to difference in Plots.*
- **- 5 pts** (b) No Plot
- **- 10 pts** No Attempt