HW2

Graded

Student

Manshi Sagar

Total Points

48 / 70 pts

Question 1

Q1

0 / 5 pts

- ✔ **+ 0 pts** Incorrect
- **+ 2.5 pts** First SVD Correct
- **+ 1.5 pts** Minor mistake in first SVD
- **+ 2.5 pts** Second SVD Correct
- **+ 1.5 pts** Minor mistake in second SVD
- **+ 0.5 pts** Correct direction in second SVD

1a : "Every matrix can be seen as a diagonal matrix" -- this is not precise enough. Work the singular values out instead of saying we can find them.

1b : Same as above. Too vague.

Question 2

Q2

8 / 15 pts

- **+ 0 pts** Incorrect
- ✔ **+ 1 pt** Part a : AV
- **+ 1 pt** Part a : UA
- ✔ **+ 1 pt** Part b
- **+ 3 pts** Part c : Correct idea
- ✔ **+ 5 pts** Part c : Correct
- **+ 3 pts** Part d : Correct idea
- **+ 5 pts** Part d : Correct
- **+ 1 pt** Part d : Use $\sigma_{k+1} \neq \sigma_k$ to infer something about B
- **+ 2 pts** Part e : Correct counterexample

+ 1 pt e : The question asks for a B st || A - B ||_2 = sigma_{k+1}, not sigma_k. Giving 1 partial mark.

Question 3

Q3

5 / 5 pts

- ✔ **+ 5 pts** Correct
- **+ 0 pts** Incorrect/Unattempted

Question 4

Q4

5 / 10 pts

- ✔ **+ 5 pts** A Correct
- **+ 5 pts** B Correct
- **+ 0 pts** Incorrect
- **− 1 pt** Late

Question 5

Q5

5 / 5 pts

- ✔ **− 0 pts** Correct
- **− 2 pts** Incomplete/Incorrect code for `formQ(W)`
- **− 3 pts** Incomplete/Incorrect code for `house(A)`
- **− 5 pts** Incorrect/Unattempted
- **− 0.5 pts** Late

Question 6

Q6

5 / 5 pts

- ✔ **− 0 pts** Correct
- **− 5 pts** Incorrect/Incomplete code for `leastSquare`
- **− 5 pts** Incorrect/Unattempted
- **− 0.5 pts** Late

Question 7

Q7

10 / 10 pts

- ✔ **− 0 pts** Correct
- **− 2 pts** Error values not shown
- **− 5 pts** random noise not added
- **− 7 pts** Only QR factorisation done
- **− 8 pts** Implementation not given, but idea is correct
- **− 10 pts** Incorrect/Unattempted
- **− 1 pt** Late

Question 8

Q8

10 / 15 pts

✔ **+ 5 pts** Least Squares Framing

✔ **+ 5 pts** Code

+ 5 pts Solution

+ 0 pts Incorrect

− 1.5 pts Late

Question assigned to the following page: 1

Q1 (a) $B = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}$

singular values don't change on row exchange
(as row exchange can be seen as multiplying by a permutation matrix P)

$B = \begin{bmatrix} A & 0 \\ 0 & A^T \end{bmatrix}$

$A = U\Sigma V^T$
$A^T = V\Sigma^T U^T$
$= V\Sigma U^T$
$\Rightarrow$ singular values of A and AT are equal

If A is a diagonal matrix,
its diagonal entries are its singular values
(we just need to rearrage the rows and columns by multiplying by a permation matrix to get the order $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$

If A is a diagonal matrix $A = \begin{bmatrix} a_1 & & & \\ & a_2 & & \\ & & \ddots & \\ & & & a_n \end{bmatrix}$, $a_1 \geq a_2 \geq \cdots \geq a_n$

$B = \begin{bmatrix} a_1 & & & & & \\ & \ddots & & & & \\ & & a_n & & & \\ & & & a_1 & & \\ & & & & \ddots & \\ & & & & & a_n \end{bmatrix}$

again we just need to rearrange the rows and columns

$B' = P_1 \begin{bmatrix} a_1 & & & & & \\ & \ddots & & & & \\ & & a_n & & & \\ & & & a_1 & & \\ & & & & \ddots & \\ & & & & & a_n \end{bmatrix} P_2$

$= \begin{bmatrix} a_1 & & & & & & \\ & a_1 & & & & & \\ & & a_2 & & & & \\ & & & a_2 & & & \\ & & & & \ddots & & \\ & & & & & a_n & \\ & & & & & & a_n \end{bmatrix}$

Hence $a_1, a_2, \ldots a_n$ are the singular values of B

Since every matrix (A) can be seen as a diagonal matrix Σ using $A = U\Sigma V^T$ using Basis conversion
we can find the singular values of $\begin{bmatrix} A & 0 \\ 0 & A^T \end{bmatrix}$
which is same as the singular values of A

Question assigned to the following page: 1

[Q1] (b) $B = \begin{pmatrix} I & A^T \\ A & 0 \end{pmatrix}$

exchanging columns of B (multiply by permutation matrix)

$B = \begin{pmatrix} A^T & I \\ 0 & A \end{pmatrix}$

$$\begin{pmatrix} I & A^{-1} \\ 0 & I \end{pmatrix} B = \begin{pmatrix} I & A^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} A^T & I \\ 0 & A \end{pmatrix}$$

$$= \begin{pmatrix} A^T & 0 \\ 0 & A \end{pmatrix}$$

we want to convert B to a diagonal matrix B'
by multiply B with an orthogonal matrix

then the singular values of B' = singular values of B

Now if A is not diagonal,
we can visualise it as a diagonal matrix using SVD (by changing basis)

Then the procedure is same as that of diagonal matrix

Question assigned to the following page: 2

Q2 ⓐ $\|AV\|_F$

$= \text{trace}\left(AV(AV)^T\right)$

$= \text{trace}\left(A\underbrace{VV^T}A^T\right)$

$= \text{trace}\left(AA^T\right)$ (V is unitary)

$= \text{trace}\left(\begin{bmatrix} — a_1 — \\ — a_2 — \\ \\ — a_m — \end{bmatrix} \begin{bmatrix} a_1 \,|\, a_2 \,|\, \cdots \,|\, a_m \end{bmatrix}\right) =$

$= (a_{11}^2 + a_{12}^2 + a_{13}^2 \cdots a_{1n}^2)$
$+ (a_{21}^2 + \cdots \quad a_{2n}^2)$
$+$
$\vdots$
$+ (a_{m1}^2 + \cdots \quad a_{mn}^2)$

$= \sum_{j=1}^{n} \sum_{i=1}^{m} (a_{ij})^2$

$= \|A\|_F^2$

Question assigned to the following page: 2

Q2 (b) $A = U\Sigma V^T$ with rank $r = \min(m, n)$

$A_k = U\Sigma_k V^T$

assuming $k < r$

$$\Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \sigma_3 & \\ & & & \ddots & \\ & & & & \sigma_r \end{bmatrix} \qquad \Sigma_k \begin{bmatrix} \sigma_1 & & & & & \\ & \sigma_2 & & & & \\ & & \ddots & & & \\ & & & \sigma_k & & \\ & & & & 0 & \\ & & & & & \ddots & \\ & & & & & & 0 \end{bmatrix}$$

$$\Sigma - \Sigma_k = \begin{bmatrix} 0 & & & & & \\ & 0 & & & & \\ & & \ddots & & & \\ & & & \sigma_{k+1} & & \\ & & & & \sigma_{k+2} & \\ & & & & & \ddots & \\ & & & & & & \sigma_r \end{bmatrix}$$

$$A - A_k = U(\Sigma - \Sigma_k)V^T$$

$$\|A - A_k\|_F = \|U(\Sigma - \Sigma_k)V^T\|_F = \|\Sigma - \Sigma_k\|_F \quad \text{(using part (a))}$$

$$= \left((\sigma_{k+1})^2 + (\sigma_{k+2})^2 + \cdots + \sigma_r^2\right)^{1/2}$$

Question assigned to the following page: 2

$\xrightarrow{\quad} m \times n \qquad m \geq n$

Q2 (c) $A = U_A \Sigma_A V^T_A \qquad A_K = U_A \, \Sigma_{AK} V^T_A \qquad$ if rank (A) < 1

$B = U_B \Sigma_B V^T_B \qquad$ rank (B) = K

let B be a better F-norm approximation of A than A_K

ie $\|A - B\|_F < |A - A_K|_F$

$\|A-B\|_F^2 < \sigma_{K+1}^2 + \dots + \sigma_r^2$ where r = Rank (A) (from part (b))

dim of nullspace(B) = $n - K$

$A = U_A \Sigma_A V_A^T \qquad B = U_B \Sigma_B V_B^T$

$A - B \longrightarrow V^{-1} U^{-1} (A - B)$

let $(N_1, N_2, \dots N_{n-k})$ be the basis of nullspace of B

$\|A - B\|_F^2 \geq \|(A-B)N_1\|_2^2 + \dots + \|(A-B)N_{n-K}\|_2^2$ —— (1)

let N' = unitary matrix w/ $N_1, \dots N_{n-K}$ as the first $(n-K)$ columns

$\|A-B\|_F^2 = \|(A-B)N'\|_F^2$

for any matrix M : $m \times n$

$\|M\|_F^2 \geq \|M_1\|_2^2 + \dots \|M_n\|_2^2$

so,

$\|(A-B)N'\|_F^2 \geq \left(\|(A-B)N_1\|_2^2 + \dots + \|(A-B)N_{n-K}\|_2^2\right)$

$\|(A-B)N_1\|_2^2 + \dots + \|(A-B)N_{n-K}\|_2^2$

$= \|AN_1\|_2^2 + \dots + \|AN_{n-K}\|_2^2$ —— (2)

let $N_1 = (N_{11}, \dots, N_{n1})$ in the new basis

then

$AN_1 = (\sigma_1 N_{11}, \dots, \sigma_n N_{n1})$

$\|AN_1\|_2^2 = \sigma_1^2 (N_{11})^2 + \dots + \sigma_n^2 (N_{n1})^2$

similarly for all $\{N_i\}$ $\qquad \|AN_i\|_2^2 = \sigma_1^2 (N_{1i})^2 + \sigma_2^2 (N_{2i})^2 + \dots + \sigma_n^2 (N_{ni})^2$

using (1) and (2)

$\|A-B\|_F^2 \geq \sigma_1^2 (N_{11}^2 + N_{12}^2 + \dots + N_{1n}^2) + \sigma_2^2 (N_{21}^2 + N_{22}^2 + \dots + N_{2n}^2)$

$+ \sigma_{n-K}^2 (\qquad) \dots + \cancel{\sigma_n^2 (N_{n1}^2 + N_{n2}^2 + \dots + N_{nn}^2)}$

Question assigned to the following page: 2

let M = matrix whose columns are $N_1 \ldots N_{n-k}$

extend M to unitary matrix

$$M' = \begin{bmatrix} N_1 & N_2 \ldots & N_{k-n} & M_{k-n+1} & \ldots & M_n \end{bmatrix}_{m \times n}$$

2 norm of row of M = 1 (M' is unitary)

$$\Rightarrow N_{11}^2 + N_{21}^2 + \ldots + (N_{k-n,1})^2 \leq 1$$

$$\Rightarrow \sigma_1^2 \left(N_{11}^2 + N_{21}^2 + \ldots (N_{k-n,1})^2\right) \leq \sigma_j^2$$

$$\sum_j \sum_i (M_{ij})^2 = n-k \geq \sum_{j=1}^{n} \sum_{i=1}^{n-k} N_{ij}^2$$

$$\| A - B \|_F^2 \geq \sigma_1^2 + \sigma_2^2 + \ldots \sigma_{n-k}^2$$

This contradicts our assumption that

$$\| A-B \|_F^2 < \sigma_{k+1}^2 + \ldots + \sigma_n^2$$

Hence proved that

$$\| A-B \|_F^2 \geq \sigma_{k+1}^2 + \ldots + \sigma_n^2$$

Question assigned to the following page: 2

Q2 (d) $\sigma_{k+1} < \sigma_k$

we know that for any B s.t. $\text{Rank}(B) = k$

$$\| A - B \|_F^2 \geq \sigma_{k+1}^2 + \cdots + \sigma_n^2$$

if $\sigma_k > \sigma_{k+1}$

ie $\sigma_k \neq \sigma_{k+1}$

let $B = U \begin{bmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_k & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{bmatrix} V^T$

$\{\sigma_i\}_{i=1}^{n}$ are singular values of A

Question assigned to the following page: 2

Q2 (c) $A = U \Sigma V^T$, $\{\sigma_i\}$ are singular values of A

$\sigma_k > \sigma_{k+1}$ holds

let $B = U \begin{bmatrix} \sigma_1+\sigma_k & & & & & \\ & \sigma_2+\sigma_k & & & & \\ & & \ddots & & & \\ & & & \sigma_k+\sigma_k & & \\ & & & & 0 & \\ & & & & & \ddots \; 0 \end{bmatrix} V^T$

$(A-B) = U \begin{bmatrix} \sigma_k & & & & & \\ & \ddots & & & & \\ & & \sigma_k & & & \\ & & & \sigma_{k+1} & & \\ & & & & \ddots & \\ & & & & & \sigma_n \end{bmatrix} V^T$

$\| A-B \|_2 = \sigma_k$

But $B \neq A_k$

Hence the statement is not true

Question assigned to the following page: 3

Q3

$B : m \times n$

$X : n \times m$

$\| BX - I \|_F$

For a matrix $A = [a_1 | a_2 | \cdots]$

$\|A\|_F^2 = \sum_j \sum_i (a_{ij})^2 = \sum_j \|a_j\|_2^2$

$= \sum (\text{2-norm of columns})^2$ —— ①

$\Rightarrow \|A\|_F$ is minimized if $\sum_j \|a_j\|_2^2$ is minimized

ie 2-norm of columns is min

$$X = \left[x_1 \middle| x_2 \middle| \cdots \middle| x_m \right]$$

$\| BX - I \|_F^2$ is minimized when

$\| Bx_i - e_i \|_2^2$ is minimized for all i

$\Rightarrow$ least squares problem

$\Rightarrow$ Solution of $\min \| BX - I \|_F^2$

= solution of least squares $\| Bx_i - e_i \|_2^2$

(for all i, the solution of least squares remains same, hence, B^+ minimizes each term in RHS of ① so, it minimizes the overall objective function)

$\Rightarrow X =$ pseudoinverse of B

$= (B^T B)^{-1} B^T$

Question assigned to the following page: 3

if $X = (B^T B)^{-1} B^T$

$$BX = B(B^T B)^{-1} B^T$$

$$= \left(B\ B^{-1}\right)\left((B^T)^{-1}\ B^T\right)$$

$$= I$$

then

$$BX - I = I - I = 0$$

Hence $\|BX - I\|_F$ will be minimized $\left(\|\ \|_F \geq 0\right)$

The minimum value of this objective function is 0.

as $\|\ \|_F \geq 0$

Question assigned to the following page: 4

Q4 ⓐ $\min \| AXB - C \|_F$

$A - BXC$

$(BXC - A)$

$A \quad B \quad C$

$$\| AXB - C \|_F^2 = (AXB - C)(AXB - C)^T$$

$$\frac{d \| AXB - C \|_F^2}{dx} = A^T (AXB - C) B^T$$

to minimise

$$A^T (AXB - C) B^T = 0$$

$$A^T AXBB^T - A^T C B^T = 0$$

$$(A^T A) \cdot X \cdot (BB^T) = A^T C B^T$$

$$= \left((A^T A)^{-1} A \right) \cdot C \cdot \left(B^T (BB^T)^{-1} \right)$$

$$\boxed{X_0 = A^+ C \cdot B^+}$$

Question assigned to the following page: 4

(b) x and x_0 both minimise $\|AXB - C\|_F$

$$\Rightarrow \quad \|AX_0B - C\|_F = \|AXB - C\|_F$$

$$\|D_1\|_F = \|D_2\|_F$$

Question assigned to the following page: 5

COL726 A2

Manshi Sagar - 2020CS50429

March 2024

1 Q5

1.1 Code:

```
function [W, R] = house(A)
    [m, n] = size(A);
    W = zeros(m, n);
    R = A;

    for k = 1:n
        x = R(k:m, k);
        vk = sign(x(1)) * norm(x) * eye(length(x), 1) + x;
        vk = vk / norm(vk);
        R(k:m, k:n) = R(k:m, k:n) - 2 * vk * (vk' * R(k:m, k:n));
        W(k:m, k) = vk;
    end
end

function Q = formQ(W)
    [m, n] = size(W);
    Q = eye(m);

    for k = n:-1:1
        vk = W(k:m, k);
        Q(k:m, :) = Q(k:m, :) - 2 * vk * (vk' * Q(k:m, :));
    end
end
```

Questions assigned to the following page: 5 and 6

```
A:
    0.6256    0.4868    0.5108    0.8116
    0.7802    0.4359    0.8176    0.5328
    0.0811    0.4468    0.7948    0.3507
    0.9294    0.3063    0.6443    0.9390
    0.7757    0.5085    0.3786    0.8759

W:
    0.8360         0         0         0
    0.2968   -0.7388         0         0
    0.0309    0.5591    0.9558         0
    0.3535   -0.3737    0.1271    0.9862
    0.2951    0.0443   -0.2652   -0.1654

R:
   -1.5723   -0.8650   -1.2176   -1.5926
    0.0000    0.4804    0.6179    0.2904
         0   -0.0000   -0.5053    0.1049
         0    0.0000         0   -0.2878
         0   -0.0000    0.0000   -0.0000

Q*R:
    0.6256    0.4868    0.5108    0.8116
    0.7802    0.4359    0.8176    0.5328
    0.0811    0.4468    0.7948    0.3507
    0.9294    0.3063    0.6443    0.9390
    0.7757    0.5085    0.3786    0.8759
```

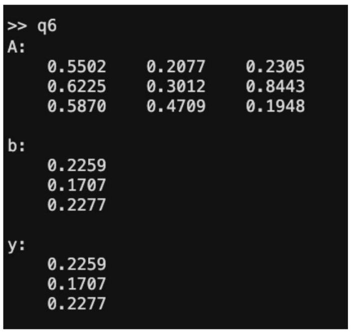
2 Q6

2.1 Code:

```
m = 3; % Number of rows
n = 3; % Number of columns
A = rand(m, n);
b = rand(m, 1);
x = leastSquare(A, b);
y = A*x;

function x = leastSquare(A, b)
    [W, R] = house(A); % Compute QR factorization using Householder reflections
    Q = formQ(W); % Reconstruct Q matrix
    % Solve Rx = Q'b using back substitution
    x = R \ (Q' * b);
end
```

Questions assigned to the following page: 6 and 7



3 Q7

3.1 Code:

```
m = 21;
n = 12;
eps = 10^(-10);
t = zeros(m, 1);
A = zeros(m, n);
y = zeros(m, 1);

for i = 1:m
  t(i) = (i-1)/(m-1);
end

for j = 0:n-1
  A(:, j+1) = t.^j;
end

for i = 1:m
  tmp = 1;
  for j = 1:n-1
    tmp = tmp + t(i)^j;
  end
  y(i) = tmp;
end

u = rand(m, 1);
y_perturbed = zeros(m, 1);
for i = 1:m
  y_perturbed(i) = y(i) + (2*u(i)-1)*eps;
end
```

Question assigned to the following page: 7

```
chol = chol_least_square(A, y);
qr = leastSquare(A, y);

chol_perturbed = chol_least_square(A, y_perturbed);
qr_perturbed = leastSquare(A, y_perturbed);

% ------------------------------------------------------------
function x = chol_least_square(A, b)
    L = chol(A'*A, 'lower');
    y = L'\(A'*b);
    x = L\y;
end
% ------------------------------------------------------------
```

```
original y:
    1.0000
    1.0526
    1.1111
    1.1765
    1.2500
    1.3333
    1.4286
    1.5385
    1.6666
    1.8181
    1.9995
    2.2205
    2.4946
    2.8409
    3.2872
    3.8733
    4.6564
    5.7184
    7.1757
    9.1928
   12.0000

y perturbed:
    1.0000
    1.0526
    1.1111
    1.1765
    1.2500
    1.3333
    1.4286
    1.5385
    1.6666
    1.8181
    1.9995
    2.2205
    2.4946
    2.8409
    3.2872
    3.8733
    4.6564
    5.7184
    7.1757
    9.1928
   12.0000
```

Question assigned to the following page: 7

```
Result of Cholesky factorization:
   1.0e+15 *

    0.0000
    0.0000
   -0.0000
    0.0000
   -0.0000
    0.0000
   -0.0001
    0.0019
   -0.0199
    0.1771
   -1.3719
    9.3353

Error in Cholesky factorization:
   9.7822e+15
```

```
Result of QR :
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000

Error in  QR factorization:
   3.5266e-15
```

```
Result of Cholesky factorization when y is perturbed:
   1.0e+15 *

    0.0000
    0.0000
   -0.0000
    0.0000
   -0.0000
    0.0000
   -0.0001
    0.0019
   -0.0199
    0.1771
   -1.3719
    9.3353

Error in  Cholesky factorization when y is perturbed:
   9.7822e+15

Result of QR when y is perturbed:
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0001
    0.9997
    1.0005
    0.9994
    1.0004
    0.9998
    1.0000

Error in QR factorization when y is perturbed:
   1.7435e-10

Condition number of A:
   1.4926e+08
```

Even without any perturbation, we can see that result of QR method is close to actual solution where as result of Cholesky factorisation has large error.
The difference in accuracy between the QR method and Cholesky factorization

Questions assigned to the following page: 7 and 8

arises from the numerical stability and conditioning of the methods. Cholesky factorization is more susceptible to numerical errors, especially in ill-conditioned systems, leading to larger errors in the solution. On the other hand, QR factorization, particularly when implemented using Householder reflections, tends to be more numerically stable and robust, resulting in solutions that are closer to the true solution even in challenging situations.
With perturbation also, we see large error when using Cholesky and insignificant error when using QR with householder.
The large error observed with Cholesky factorization, even with perturbation, is due to its sensitivity to numerical errors and ill-conditioned systems. QR factorization with Householder reflections, on the other hand, maintains numerical stability, leading to insignificant errors even in the presence of perturbations.
We also see that the condition number of matrix A is very large. A large condition number of matrix A indicates that the matrix is ill-conditioned. So, we can conclude that Cholesky does not work well with ill-conditioned matrices whereas QR is very stable even with ill-conditioned matrices.
Comparing the results before and after perturbation, we see that QR is more sensitive to perturbation than Cholesky.
To recover the x we generated, QR factorization comes closer as it is backward stable.

4 Q8

To formulate this problem as a least squares problem, we want to minimize the discrepancy between the measured distances and the actual distances between the points. Let (x_i, y_i) be the original coordinates of the i-th point, and let δ_i and ϵ_i be the corrections to the x-coordinate and y-coordinate of the i-th point, respectively. We want to find the corrections δ_i and ϵ_i that minimize the squared discrepancy between the measured distances and the actual distances.

We start by expressing the squared Euclidean distance between two points i and j as a linear expression in the unknowns δ_i and ϵ_i. Expanding the squared distance and removing terms involving the product or square of δ_i and ϵ_i, we get:

$$(x_i + \delta_i - x_j - \delta_j)^2 + (y_i + \epsilon_i - y_j - \epsilon_j)^2$$

$$= (x_i - x_j)^2 + 2(x_i - x_j)(\delta_i - \delta_j) + (\delta_i - \delta_j)^2 + (y_i - y_j)^2 + 2(y_i - y_j)(\epsilon_i - \epsilon_j) + (\epsilon_i - \epsilon_j)^2$$

$$= (x_i - x_j)^2 + (\delta_i - \delta_j)^2 + (y_i - y_j)^2 + (\epsilon_i - \epsilon_j)^2 + 2(x_i - x_j)(\delta_i - \delta_j) + 2(y_i - y_j)(\epsilon_i - \epsilon_j)$$

$$= d_{ij}^2 + (\delta_i - \delta_j)^2 + (\epsilon_i - \epsilon_j)^2 + 2(x_i - x_j)(\delta_i - \delta_j) + 2(y_i - y_j)(\epsilon_i - \epsilon_j)$$

$$= d_{ij}^2 + 2(x_i - x_j)(\delta_i - \delta_j) + 2(y_i - y_j)(\epsilon_i - \epsilon_j)$$

Here, d_{ij} represents the measured distance between points i and j.

Let D_{ij} be the matrix containing the measured pair-wise distances. We can construct a system of equations where each equation represents the above linear

Question assigned to the following page: 8

expression for each pair of distinct points. We want to minimize the sum of squared discrepancies:

$$minimize \sum_{i<j} \left(d_{ij}^2 + (\delta_i - \delta_j)^2 + (\epsilon_i - \epsilon_j)^2 + 2(x_i - x_j)(\delta_i - \delta_j) + 2(y_i - y_j)(\epsilon_i - \epsilon_j) - D_{ij}^2\right)^2$$

$$min \sum_{i<j} \left(d_{ij}^2 + 2(x_i - x_j)(\delta_i - \delta_j) + 2(y_i - y_j)(\epsilon_i - \epsilon_j) - D_{ij}^2\right)^2$$

$$min \sum_{i<j} \left(d_{ij}^2 + 2(x_i - x_j)(\delta_i) - 2(x_i - x_j)(\delta_j) + 2(y_i - y_j)(\epsilon_i) - 2(y_i - y_j)(\epsilon_j) - D_{ij}^2\right)^2$$

We want to find the values of δ_i and ϵ_i that minimize this objective function, subject to the constraints.

We can solve this system of equations using least squares techniques to find the corrections δ_i and ϵ_i to the original points (x_i, y_i). These corrections will give us the more precise coordinates of the points that minimize the discrepancy between the measured distances and the actual distances.
x = $(\delta_1, \epsilon_1, \delta_2, \epsilon_2, \delta_3, \epsilon_3, \delta_4, \epsilon_4, \delta_5, \epsilon_5, \delta_6, \epsilon_6)$

4.1 Code:

```
x = [1, 2, 3, 4, 1.5, 3.4];
y = [2, 4, 5, 6, 5.5, 2.7];
b = [4.674, 11.349, 25.632, 13.075, 5.806, 1.981, 9.360, 3.009, 4.036, 2.955, 4.064, 4.105,

distance_matrix = zeros(6, 6);

for i = 1:6
    for j = i+1:6
        distance_matrix(i, j) = (x(j) - x(i))^2 + (y(j) - y(i))^2;
    end
end

newd = [];

for i = 1:size(distance_matrix, 1)
    for j = i+1:size(distance_matrix, 2)
        if distance_matrix(i, j) > 0
            newd = [newd, distance_matrix(i, j)];
        end
    end
end
for i = 1:numel(b)
    b(i) = b(i) - newd(i);
end
```

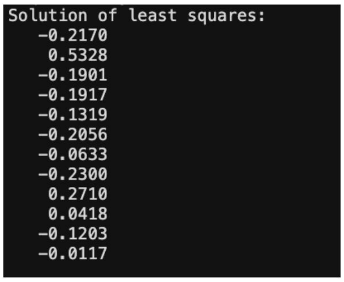
Question assigned to the following page: 8

```
disp('Vector b:');
disp(b);

A = zeros(15, 12);

for i = 1:5
    for j = i+1:6
         A(i, i) = x(i) - x(j);
         A(i, i+6) = y(i) - y(j);
         A(i, j) = -(x(i) - x(j));
         A(i, j+6) = -(y(i) - y(j));
    end
end

disp('Matrix A:');
disp(A);
x = lsqr(A, b.');
disp(x);
```



```
Solution of least squares:
   -0.2170
    0.5328
   -0.1901
   -0.1917
   -0.1319
   -0.2056
   -0.0633
   -0.2300
    0.2710
    0.0418
   -0.1203
   -0.0117
```