תרגיל בית 1 – מודלים לא ליניאריים בחקר ביצועים

מגישים :

אלעד בוכריס – 206202426

משה דידי – 311395834

פונקציית בדיקת הקלט עבור שאלות 1 ו2:

```python
def is_valid_input(A=None, B=None, x=None):
    valid, error_message = True, ''
    n, m = 0, 0
    if A is not None:
        try:
            # Try to create a numpy array_like from given A lists
            A = np.array(A)
            # Validate that all matrix elements are numbers
            _type = np.array(list(A)).dtype
            if _type not in ['int32', 'float64']:
                valid = False
                error_message += ' Matrix A is not all numbers,'
            else:
                # Validate that the matrix is squared and has at least one element
                n_temp, m_temp = A.shape
                if n_temp != m_temp or n_temp <= 0:
                    valid = False
                    error_message += ' Matrix A dimensions are not valid,'
                else:
                    n, m = n_temp, m_temp
        except Exception as err:
            valid = False
            error_message += f' {str(err)},'
    if B is not None:
        try:
            # Try to create a numpy array_like from given B lists if B is given
            B = np.array(B)
            # Validate that all matrix elements are numbers
            _type = np.array(list(B)).dtype
            if _type not in ['int32', 'float64']:
                valid = False
                error_message += ' Matrix B is not all numbers,'
            else:
                # Validate that the matrix is squared and has at least one element and it's
dimensions are the same of A
                n_temp, m_temp = B.shape
                if n_temp != m_temp or n_temp <= 0 or n_temp != n or m_temp != m:
                    valid = False
                    error_message += ' Matrix B dimensions are not valid,'
        except Exception as err:
            valid = False
            error_message += f' {str(err)},'
    if x is not None:
        try:
            # Try to create a numpy array from given x
            x = np.array(x)
            # Validate that all array elements are numbers
            _type = np.array(list(x)).dtype
            if _type not in ['int32', 'float64']:
                valid = False
                error_message += ' Array x is not all numbers,'
            else:
                # Validate that the array dimensions are the appropriate to A
                n_temp = x.shape[0]
                if n_temp != n or n_temp <= 0 or len(x.shape) != 1:
                    valid = False
                    error_message += ' Array x dimensions are not valid,'
        except Exception as err:
            valid = False
            error_message += f' {str(err)},'
    if error_message != '':
        print(error_message)
    return valid
```

שאלה 1 :

```python
def ex1(A, x):
    if not is_valid_input(A=A, x=x):
        return
    n = A.shape[0]
    X = np.array([x, ] * n).T
    i = np.arange(1, n + 1)
    B = np.add(A, np.multiply(X, i)).T
    np.fill_diagonal(B, 0)
    return B
```

פלט :

```
EX1:
[[  0  10 -10  10]
 [ 32   0   2 -11]
 [ 54  24   0  12]
 [ 75  31  -5   0]]
```

שאלה 2 :

```python
def ex2(A, B, n, b):
    if not is_valid_input(A=A, B=B, x=b):
        return
    """ Create P """
    BABT = np.block([[B, A, B.T]])
    m = A.shape[0]
    P = np.block([[A, B.T, np.array([np.zeros(m), ] * (n - 2) * m).T]])
    second = np.block([[BABT, np.array([np.zeros(m), ] * (n - 3) * m).T]])
    P = np.append(P, second, axis=0)
    for i in range(1, n - 3):
        temp = np.block([[np.array([np.zeros(m), ] * i * m).T, BABT,
np.array([np.zeros(m), ] * (n - i - 3) * m).T]])
        P = np.append(P, temp, axis=0)
    before_last = np.block([[np.array([np.zeros(m), ] * (n - 3) * m).T, BABT]])
    P = np.append(P, before_last, axis=0)
    last = np.block([[np.array([np.zeros(m), ] * (n - 2) * m).T, B, A]])
    P = np.append(P, last, axis=0)
    """ Create y """
    y = b.T
    for i in range(2, n + 1):
        temp = b * i
        y = np.append(y, temp, axis=0)
    """ Create Q """
    Q = np.kron(A, P)
    """ Create z """
    z = np.block([[y] * m])
    return np.linalg.solve(Q, z.T)
```

פלט :

```
EX2:
[[ 6.51495099e+00]
 [-5.93145486e+01]
 [ 6.40000000e+01]
 [-9.02260826e+01]
 [ 2.60823821e+02]
 [-1.44000000e+02]
 [-5.62972627e+00]
 [-1.85299306e+01]
 [-1.74062500e+01]
 [ 1.57235343e+01]
 [-7.35019720e+01]
 [ 4.59826095e+01]
 [ 1.01789869e+01]
 [ 6.87798599e+00]
 [-3.96800000e+01]
 [-6.29707739e+01]
 [-1.39128759e+01]
 [ 2.33548387e+01]
 [ 3.26383106e+01]
 [ 8.07992284e+00]
 [ 4.20802469e+01]
 [ 2.44500366e+01]
 [ 2.07652889e+01]
 [-2.26236693e+01]
 [-1.37492061e-01]
 [ 1.93236709e+01]
 [-7.76355416e+00]
 [ 2.92435384e+01]
 [ 9.95690550e-01]
 [-3.30815675e+00]
 [-1.39928627e+01]
 [-1.55814355e+01]
 [-1.16582753e+01]
 [-8.33689620e+00]
 [-1.09366665e+01]
 [ 8.47773453e+00]]
```

שאלה 3 :

א.

טענה 1 :

$$argmax\ \{f(x): x \in C\} = argmin\ \{-f(x): x \in C\}$$

הוכחה :

נסמן

$$x^* \in argmax\ \{f(x): x \in C\} \rightarrow f(x^*) \geq f(x)\ \forall x \in C \rightarrow -f(x^*) \leq -f(x)\ \forall x \in C$$
$$\rightarrow x^* \in argmin\ \{-f(x): x \in C\}$$

מש״ל

טענה 2 :

$$\max\{\alpha f(x) : x \in C\} = \alpha \max\{f(x) : x \in C\}\ \ \forall \alpha \geq 0$$

נסמן :

$$M = \max\{f(x) : x \in C\} \rightarrow M \geq f(x)\ \forall x \in C \rightarrow \alpha M \geq \alpha f(x)\ \forall x \in C, \alpha \geq 0$$
$$\rightarrow \alpha M = \max\{g(x): x \in C, g(x) = \alpha f(x), \alpha \geq 0\} \rightarrow \alpha M = \max\{\alpha f(x) : x \in C\}$$

מש״ל

טענה 3 :

$$argmax\{f(x) + \alpha: x \in C\} = argmax\ \{f(x): x \in C\}\ \ \forall \alpha \in \mathbb{R}$$

נסמן :

$$x^* \in argmax\ \{f(x): x \in C\}$$
$$\rightarrow f(x^*) \geq f(x)\ \forall x \in C \rightarrow f(x^*) + \alpha \geq f(x) + \alpha\ \forall x \in C \land \forall \alpha \in \mathbb{R}$$
$$\rightarrow x^* = \mathrm{argmax}\{g(x): x \in C, g(x) = f(x) + \alpha, \alpha \in \mathbb{R}\}$$
$$\rightarrow x^* \in argmax\{f(x) + \alpha : x \in C, \alpha \in \mathbb{R}\}$$

מש״ל

טענה 4 :

$$\max\{f(x) + g(x): x \in C\} \leq \max\{f(x): x \in C\} + \max\{g(x): x \in C\}$$

הוכחה :

נסמן

$$F = \max\{f(x): x \in C\}$$
$$G = \max\{g(x): x \in C\}$$

ולכן :

$$(G \geq g(x)\ \forall x \in C) \land (F \geq f(x)\ \forall x \in C)$$
$$\rightarrow G + F \geq g(x) + f(x)\ \forall x \in C \rightarrow G + F \geq \max\{f(x) + g(x): x \in C\}$$

מש״ל

טענה 5 :

בהינתן $D \subseteq C$ אזי :

$$\max\{f(x): x \in C\} \leq \max\{f(x): x \in D\}$$
$$\min\{f(x): x \in C\} \geq \min\{f(x): x \in D\}$$

נסמן :

$$x^* \in \text{argmax}\{f(x): x \in C\} \rightarrow x^* \in C \rightarrow x^* \in D$$

לכן, ניתן לראות כי לכל הפחות הערך המקסימלי תחת קבוצה D יתקבל תחת ה-x הנותן את הערך המקסימלי תחת C (מכיוון שאנו ממקסמים).
בנוסף, ייתכן ש-

$$\exists x^{**} \in D \setminus C : f(x^{**}) > f(x^*)$$

ולכן אי השוויון מתקיים.
באופן דומה עבור המינימום.

ב. תהי

$$f: C \subseteq \mathbb{R}^n \rightarrow \mathbb{R} \ : \exists x^* = argmax\{f(x): x \in C\}$$

יש להוכיח כי אם

$$\exists x_* = argmin\{f(x): x \in C\}$$

כך ש-

$$f(x_*) > 0$$

אזי :

$$argmax\{f(x): x \in C\} = argmin\left\{\frac{1}{f(x)}: x \in C\right\}$$

לפי הנתון מתקיים כי

$$1. f(x_*) > 0 \rightarrow f(x) > 0 \ \forall x \in C$$
$$x^* = argmax\{f(x): x \in C\} \rightarrow f(x^*) \geq f(x) \ \forall x \in C$$
$$\xrightarrow{from\ 1} \frac{1}{f(x^*)} \leq \frac{1}{f(x)} \ \forall x \in C \rightarrow x^* = argmin\left\{\frac{1}{f(x)}: x \in C\right\}$$

מש״ל.

הטענה אינה נכונה ללא ההנחה.
דוגמה נגדית :

$$f(x) = x : C = [-1,1]$$

ניתן לראות כי ההנחה לא מתקיימת מכיוון ש-

$$f(argmin\{f(x): x \in C\}) = f(-1) = -1 < 0$$

בנוסף, מתקיים כי :

$$argmax\{f(x): x \in C\} = 1 \neq argmin\left\{\frac{1}{f(x)}: x \in C\right\} = -1$$

לכן, הטענה אינה נכונה.

שאלה 4 :

יש לפתור את הבעיה הבאה :

$$\underset{x\in\mathbb{R}^n}{max}\{x^TAx : \|x\| = 1\}$$

עבור מטריצה $A \in \mathbb{R}^{n\times n}$ סימטרית.

ננסה למצוא חסם עליון :

מכיוון שהמטריצה סימטרית ממשית אזי קיימת מטריצה U כך ש-

$$A = Udiag(\lambda_1, .., \lambda_n)U^T$$

כאשר λ_i ע״ע ממשי של A.

ולכן :

$$x^TAx = x^TUdiag(\lambda_1, .., \lambda_n)U^Tx$$

נפתח המכפלה :

$$(x_1, x_2, \dots, x_n)\cdot\begin{pmatrix} u_{11} & \cdots & u_{1n} \\ \vdots & \ddots & \vdots \\ u_{n1} & \cdots & u_{nn}\end{pmatrix} diag(\lambda_1, .., \lambda_n)\begin{pmatrix} u_{11} & \cdots & u_{n1} \\ \vdots & \ddots & \vdots \\ u_{1n} & \cdots & u_{nn}\end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n\end{pmatrix} =$$

$$\left(\sum_{i=1}^n x_iu_{i1}, \quad \cdots \quad, \sum_{i=1}^n x_iu_{in}\right) diag(\lambda_1, .., \lambda_n)\begin{pmatrix} \sum_{j=1}^n x_ju_{1j} \\ \vdots \\ \sum_{j=1}^n x_ju_{nj}\end{pmatrix} =$$

$$\left(\sum_{i=1}^n x_iu_{i1}, \quad \cdots \quad, \sum_{i=1}^n x_iu_{in}\right)\begin{pmatrix} \lambda_1\sum_{j=1}^n x_ju_{1j} \\ \vdots \\ \lambda_n\sum_{j=1}^n x_ju_{nj}\end{pmatrix} \overset{symetry}{=} \sum_{k=1}^n \lambda_k\left(\sum_{i=1}^n x_iu_{ik}\right)^2$$

$$\le \sum_{k=1}^n \lambda_{max}\left(\sum_{i=1}^n x_iu_{ik}\right)^2 = \lambda_{max}\sum_{k=1}^n\left(\sum_{i=1}^n x_iu_{ik}\right)^2$$

$$= \lambda_{max}\sum_{k=1}^n\left(\sum_{i=1}^n x_i^2u_{ik}^2 + 2\sum_{j=1}^n\sum_{m=1}^{j-1} x_ju_{jk}x_mu_{mk}\right) =$$

$$\lambda_{max}\left[\sum_{i=1}^n\sum_{k=1}^n x_i^2u_{ik}^2 + 2\sum_{j=1}^n\sum_{m=1}^{j-1}\sum_{k=1}^n x_ju_{jk}x_mu_{mk}\right]$$

נשים לב כי :

$$\sum_{i=1}^n\sum_{k=1}^n x_i^2u_{ik}^2 \overset{change\ sum\ order}{=} \sum_{i=1}^n x_i^2\sum_{k=1}^n u_{ik}^2 \overset{a}{=} \|x\|_2^2 = 1$$

וכי:

$$2\sum_{j=2}^{n}\sum_{m=1}^{j-1}x_j x_m \sum_{k=1}^{n}u_{jk}u_{mk} \overset{symerty}{=} 2\sum_{j=2}^{n}\sum_{m=1}^{j-1}x_j x_m \sum_{k=1}^{n}u_{jk}u_{km} \overset{b}{=} 0$$

ולכן מתקיים כי

$$x^T U diag(\lambda_1, .. , \lambda_n) U^T x \leq \lambda_{max}$$

כעת, נמצא ווקטור x כך שהחסם העליון מתקבל.
נבחר את x להיות הו״ע העצמי המתאים לערך העצמי המקסימלי.
נסמנו ב- x^* אזי:

$$\boldsymbol{x^{*T} A x^* = x^{*T}\lambda_{max} x^* = \lambda_{max}\|x^*\|^2 \overset{\|x^*\|=1}{=} \lambda_{max}}$$

ניתן לראות כי החסם העליון מתקבל וזהו הערך האופטימלי לבעיה.

a. המטריצה U היא אורתונורמלית, לכן, $\|u\|^2 = 1\ \forall u \in U_{columns}$.

b. מאותה סיבה, היא אורתונורמלית ולכן המכפלה הסקלרית של כל שתי עמודות שונות ב-U היא אפס.

שאלה 5 :

נתונות $\{(x_i,y_i)\}_{i=1}^{m}$ הניתנות לקירוב על ידי :

$$y_i \approx f(x_i) = \frac{c_0 + c_1x_i + c_2x_i^2}{d_0 + d_1x_i + d_2x_i^2}$$

א. נניח כי $d_0 = 1$ על מנת להתגבר על היתירות. ולכן $u = (c_0, c_1, c_2, d_1, d_2)$:
יש למצוא מטריצה A וקטור b כך שבעיית הריבועים הפחותים שלנו תהיה :

$$\min_u \|Au - b\|_2^2$$

ניתן לראות כי

$$f(x_i) - y_i \approx 0 \ \forall i \in [m]$$

ולכן :

$$\left(\frac{c_0 + c_1x_i + c_2x_i^2}{1 + d_1x_i + d_2x_i^2} - y_i\right)^2 \approx 0$$

נשים לב כי :

$$\left(\frac{c_0 + c_1x_i + c_2x_i^2}{1 + d_1x_i + d_2x_i^2} - y_i\right)^2 = \left(\frac{c_0 + c_1x_i + c_2x_i^2 - y_i(1 + d_1x_i + d_2x_i^2)}{1 + d_1x_i + d_2x_i^2}\right)^2 \approx 0 \to$$
$$c_0 + c_1x_i + c_2x_i^2 - y_i(1 + d_1x_i + d_2x_i^2) \approx 0$$

אנו רוצים למזער ביטוי זה לכל i, לכן נמזער את הסכום :

$$\min_u \sum_{i=1}^{m} \left(c_0 + c_1x_i + c_2x_i^2 - y_i(1 + d_1x_i + d_2x_i^2)\right)^2$$

נשים לב כי :

$$\sum_{i=1}^{m} \left(c_0 + c_1x_i + c_2x_i^2 - y_i(1 + d_1x_i + d_2x_i^2)\right)^2 =$$

$$= \left\| \begin{pmatrix} 1 & x_1 & x_1^2 & -y_1x_1 & -y_1x_1^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_m & x_m^2 & -y_mx_m & -y_mx_m^2 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ d_1 \\ d_2 \end{pmatrix} - \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} \right\|^2$$

לכן : הבעיה היא :

$$\min_u \left\| \begin{pmatrix} 1 & x_1 & x_1^2 & -y_1x_1 & -y_1x_1^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_m & x_m^2 & -y_mx_m & -y_mx_m^2 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ d_1 \\ d_2 \end{pmatrix} - \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} \right\|^2$$

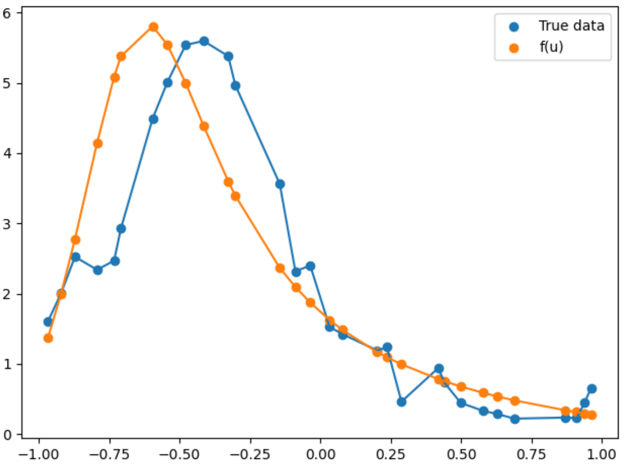
ב.

```
# ex_5_b
def fit_rational(X):
    m = X.shape[0]
    A, y = create_A_and_y(X, m)
    x = np.linalg.lstsq(A, y, rcond=None)
    return x
```

```python
def create_A_and_y(X, m):
    y = X[:, 1]
    A = np.zeros((m, 5))
    A[:, 0] = np.ones(m)
    A[:, 1] = X[:, 0]
    A[:, 2] = np.square(X[:, 0])
    A[:, 3] = -np.multiply(X[:, 0], y)
    A[:, 4] = -np.multiply(np.square(X[:, 0]), y)
    return A, y
```

פלט :

```
u is : [ 1.74018225  0.49321152 -0.93418242  2.34611649  1.6613219 ]
```



ג. כעת, נניח כי $\|u\| = 1$

ניתן לייצג את הבעיה כעת בצורה הבאה :

נכתוב באופן דומה למה שקיבלנו בסעיף הקודם

$$\min_u \|Au\|^2 = \min_u \left\| \begin{pmatrix} 1 & x_1 & x_1^2 & -y_1 & -y_1x_1 & -y_1x_1^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_m & x_m^2 & -y_m & -y_mx_m & -y_mx_m^2 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ d_0 \\ d_1 \\ d_2 \end{pmatrix} \right\|^2 : \|u\| = 1$$

נשים לב כי

$$\|Au\|^2 = u^T A^T A u = u^T (A^T A) u$$

כלומר, אנו רוצים לפתור את הבעיה הבאה :

$$\min_u u^T (A^T A) u : \|u\| = 1$$

נשים לב כי $A^T A$ תמיד מטריצה סימטרית ולכן מקבלים בעיה דומה לשאלה 4.

בשאלה 4 ראינו כי הערך המקסימלי של הבעיה הוא הע״ע המקסימלי, באופן דומה, הערך המינימלי של הבעיה הוא הע״ע המינימלי.

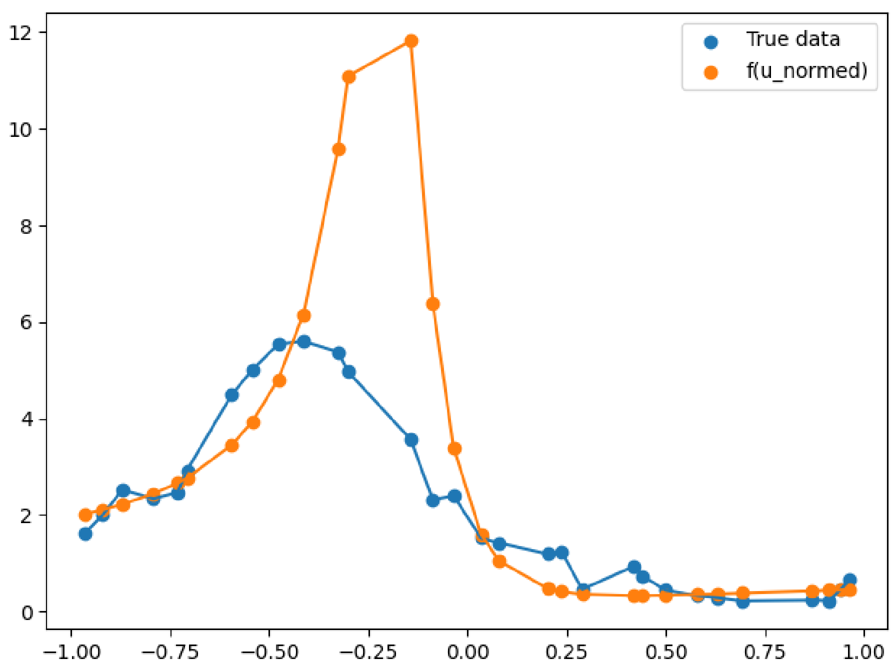
ולכן u צריך להיות הו״ע המתאים לערך העצמי המינימלי.

ד.

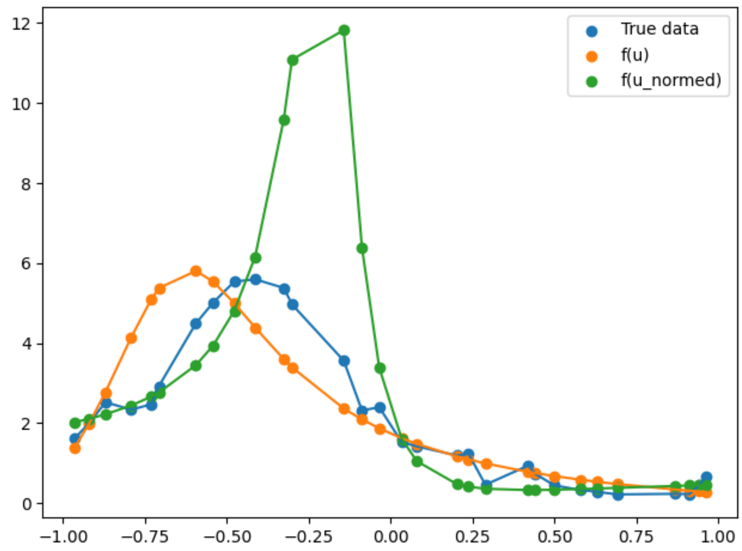
```python
# ex_5_d
def fit_rational_normed(X):
    m = X.shape[0]
    ATA = create_A_TA(X, m)
    i = np.argmin(np.linalg.eig(ATA)[0])
    x = np.linalg.eig(ATA)[1][:, i]
    return x

def create_A_TA(X, m):
    y = X[:, 1]
    A = np.zeros((m, 6))
    A[:, 0] = np.ones(m)
    A[:, 1] = X[:, 0]
    A[:, 2] = np.square(X[:, 0])
    A[:, 3] = -y
    A[:, 4] = -np.multiply(X[:, 0], y)
    A[:, 5] = -np.multiply(np.square(X[:, 0]), y)
    ATA = np.dot(A.T, A)
    return ATA
```

פלט:

```
u normed is : [ 0.07613583 -0.23284937  0.61771818  0.03340038  0.26105315  0.69938861]
```



בגרף הבא מוצגים הנתונים, הקירוב לפי u והקירוב לפי u_norm :



בנוסף נחשב את נורמת המרחק של הנתונים והעקומות הקירוב :

```
The norm of the distance: 5.306125150881595
The norm of the distance from u_norm: 12.10536847887759
```

לכן ניתן לראות כי המרחק בין נתונים והקירוב לפי סעיף א קטן מהמרחק לפי סעיף ג.

גם עפ״י הגרף ניתן לראות כי המגמה תואמת יותר לנתונים לפי הקירוב מסעיף א לכן נסיק כי הגישה לפי סעיף א׳ נותנת קירוב טוב יותר.