<u>קורס אנליזה נומרית – עבודת בית מס' 4</u>

שאלה 1:

צריך להוכיח שמתקיים $\|x-y\| \geq |\|x\|-\|y\||$ כאשר $x, y \in R^n$.

נזכר בתכונות נורמה:

- הומוגניות: $\|cx\| = |c|\|x\|$ לכל $c \in R$.
- אי שוויון המשולש: $\|x+y\| \leq \|x\|+\|y\|$.

נשתמש בתכונות הללו במהלך ההוכחה:

בהינתן $x, y \in R^n$ מתקיים:

$$\|y\| = \|y-x+x\| = \|(y-x)+x\|$$

מאי שוויון המשולש מתקיים:

$$\|y\| = \|(y-x)+x\| \leq \|y-x\|+\|x\|$$

נעביר אגפים ונקבל את ביטוי א':

$$\|y-x\| \geq \|y\|-\|x\|$$

באופן דומה, ע"י החלפת x, y בחישוב הנ"ל ונקבל את ביטוי ב':

$$\|\boldsymbol{x}-\boldsymbol{y}\| \geq \|\boldsymbol{x}\|-\|\boldsymbol{y}\|$$

מהומוגניות, עבור ביטוי א' מתקיים:

$$\|y-x\| = \|x-y\| \geq \|y\|-\|x\| = -(\|x\|-\|y\|)$$

כלומר,

$$\|\boldsymbol{x}-\boldsymbol{y}\| \geq -(\|\boldsymbol{x}\|-\|\boldsymbol{y}\|)$$

ובסה"כ, משני הביטויים ומהגדרת ערך מוחלט קיבלנו:

$$\|\boldsymbol{x}-\boldsymbol{y}\| \geq |\|\boldsymbol{x}\|-\|\boldsymbol{y}\||$$

מ.ש.ל.

שאלה 2:

צריך להוכיח שמתקיים $\|AB\| \leq \|A\| \cdot \|B\|$ כאשר $A, B \in R^n \times R^n$.

בהינתן מטריצה $A \in R^n \times R^n$

1. הגדרנו בכיתה שמתקיים $\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}$ כאשר $x \in R^n$.
2. מהגדרה זו נובע $\|A\| \geq \frac{\|Ax\|}{\|x\|}$.

נשתמש בתכונות הללו במהלך ההוכחה:

בהינתן $A, B \in R^n \times R^n$ מ- 1 נובע:

$$\|AB\| = \sup_{x \neq 0} \frac{\|ABx\|}{\|x\|} = \sup_{\substack{x \neq 0 \\ Bx \neq 0}} \frac{\|ABx\| \cdot \|Bx\|}{\|x\| \cdot \|Bx\|} = \sup_{\substack{x \neq 0 \\ Bx \neq 0}} \left(\frac{\|A(Bx)\|}{\|Bx\|} \cdot \frac{\|Bx\|}{\|x\|} \right)$$

מ- 2 נובע:

$$\frac{\|A(Bx)\|}{\|Bx\|} \cdot \frac{\|Bx\|}{\|x\|} \leq \|A\| \cdot \|B\|$$

נציב בביטוי המקורי:

$$\|AB\| = \sup_{\substack{x \neq 0 \\ Bx \neq 0}} \left(\frac{\|A(Bx)\|}{\|Bx\|} \cdot \frac{\|Bx\|}{\|x\|} \right) \leq \sup_{\substack{x \neq 0 \\ Bx \neq 0}} \|A\| \cdot \|B\| = \|A\| \cdot \|B\|$$

ובסה״כ קיבלנו:

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

מ.ש.ל.

שאלה 3:

בהינתן $A = \begin{pmatrix} 9.7 & 6.6 \\ 4.1 & 2.8 \end{pmatrix}$, $x = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $b = \begin{pmatrix} 9.7 \\ 4.1 \end{pmatrix}$, כאשר המערכת הלינארית הינה $Ax = b$,

נחשב את $cond(A)$.

<u>ראשית, נעריך את סדר הגודל של ערכו:</u>

הוכחנו בתרגול שמתקיים $\frac{\lambda_{max}}{\lambda_{min}} \leq cond(A)$.

חישבנו ומצאנו:

$$\lambda_{max} = \frac{125+\sqrt{15585}}{20}, \lambda_{min} = \frac{125-\sqrt{15585}}{20}$$

נציב:

$$\boldsymbol{cond(A)} \geq \frac{\frac{125+\sqrt{15585}}{20}}{\frac{125-\sqrt{15585}}{20}} = \frac{125+\sqrt{15585}}{125-\sqrt{15585}} = \mathbf{1560.49935}$$

מכך אנו מקבלים הערכה ש- $cond(A)$ הינו מספר גדול.

<u>שנית, נחשב את ערכו המדויק:</u>

הגדרנו בכיתה שמתקיים $cond(A) = \|A\| \cdot \|A^{-1}\|$.

נמצא את A^{-1}:

$$\left(\begin{matrix} 9.7 & 6.6 \\ 4.1 & 2.8 \end{matrix}\middle|\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}\right) \to \left(\begin{matrix} 9.7 & 6.6 \\ 0 & 0.0103 \end{matrix}\middle|\begin{matrix} 1 & 0 \\ -0.42268 & 1 \end{matrix}\right)$$
$$\to \left(\begin{matrix} 1 & 0.68041 \\ 0 & 1 \end{matrix}\middle|\begin{matrix} 0.10309 & 0 \\ -41.03689 & 97.08737 \end{matrix}\right)$$
$$\to \left(\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}\middle|\begin{matrix} 28.025 & -66.05921 \\ -41.03689 & 97.08737 \end{matrix}\right)$$

קיבלנו: $A^{-1} = \begin{pmatrix} 28.025 & -66.05921 \\ -41.03689 & 97.08737 \end{pmatrix}$.

ההגדרה אינה מגדירה נורמה מסוימת ולכן נדגים את החישוב באמצעות שתי נורמות ידועות:

- עבור $column - sum\ norm$ המוגדרת $\|A\|_1 = \max_k \sum_{i=1}^{n} |a_{ik}|$.

מתקיים:

$$\|A\|_1 = 9.7 + 4.1 = 13.8$$

$$\|A^{-1}\|_1 = 66.05921 + 97.08737 = 163.14658$$

נציב:

$$\boldsymbol{cond(A)} = 13.8 \cdot 163.14658 = \mathbf{2251.422804}$$

- עבור $row - sum\ norm$ המוגדרת $\|A\|_\infty = \max_i \sum_{k=1}^{n} |a_{ik}|$.

מתקיים:

$$\|A\|_{\infty} = 9.7 + 6.6 = 16.3$$

$$\|A^{-1}\|_{\infty} = 41.03689 + 97.08737 = 138.12426$$

נציב:

$$\boldsymbol{cond(A)} = 16.3 \cdot 138.12426 = \mathbf{2251.425438}$$

שאלה 4:

בהינתן מערכת המשוואות הבאה:

$$\begin{cases} 4x_1 - x_2 + x_3 = 7 \\ 4x_1 - 8x_2 + x_3 = -21 \\ -2x_1 + x_2 + 5x_3 = 15 \end{cases}$$

נפתור אותה בשיטת $Jacobi$ ובשיטת $Gauss\ Seidel$,

כאשר הניחוש ההתחלתי הינו $x^0 = (1, 2, 2)$.

נשים לב שכאשר מערכת המשוואות הנ"ל כתובה במטריצה הינה $diagonally\ dominant$ ולכן ביצוע צעד זה מיותר בדוגמה הזו ואינו בא לידי ביטוי בחישובים הבאים.

נציג תוכנית פיתון המבצעת את החישוב על פי הנוסחאות הבאות:

עבור שיטת $Jacobi$:

$$x^{t+1} = D^{-1}b - D^{-1}(L + U)x^t$$

עבור שיטת $Gauss\ Seidel$:

$$x^{t+1} = (L + D)^{-1}b - (L + D)^{-1}Ux^t$$

להלן התוכנית:

```python
def Jacobi(A, b, x, iter):
    [L, D, U] = LDU(A)
    G = np.matmul(np.negative(inv(D)), np.add(L, U))  # -(D^-1)*(L+U)
    C = np.matmul(inv(D), b)  # (D^-1)*b

    Jac = np.zeros((iter, 3))
    Jac[0] = x

    for i in range(1, iter):
        Jac[i] = np.add(np.matmul(G, Jac[i - 1]), C)

    return Jac


def GaussSeidel(A, b, x, iter):
    [L, D, U] = LDU(A)
    G = np.matmul(np.negative(inv(np.add(L, D))), U)  # -((L+D)^-1)*U
    C = np.matmul(inv(np.add(L, D)), b)  # ((L+D)^-1)*b

    GS = np.zeros((iter, 3))
    GS[0] = x

    for i in range(1, iter):
        GS[i] = np.add(np.matmul(G, GS[i - 1]), C)

    return GS
def LDU(A):
    return np.tril(A, -1), np.diag(np.diag(A)), np.triu(A, 1)


def print_tables(Jac, GS):
    headers = ['x1', 'x2', 'x3']
    print(f'Jacobi\n{tabulate(Jac, headers, tablefmt="fancy_grid")}')
    print(f'GaussSeidel\n{tabulate(GS, headers, tablefmt="fancy_grid")}')


def main():
    A = np.array([[4, -1, 1], [4, -8, 1], [-2, 1, 5]])  # A is diagonally dominant
    b = np.array([7, -21, 15])
    x = np.array([1, 2, 2])
    iter = 13
    Jac = Jacobi(A, b, x, iter)
    GS = GaussSeidel(A, b, x, iter)
    print_tables(Jac, GS)


if __name__ == "__main__":
    main()
```

פלט התוכנית הינו :

Jacobi

x1	x2	x3
1	2	2
1.75	3.375	3
1.84375	3.875	3.025
1.9625	3.925	2.9625
1.99063	3.97656	3
1.99414	3.99531	3.00094
1.99859	3.99719	2.99859
1.99965	3.99912	3
1.99978	3.99982	3.00004
1.99995	3.99989	2.99995
1.99999	3.99997	3
1.99999	3.99999	3
2	4	3

GaussSeidel

x1	x2	x3
1	2	2
1.75	3.75	2.95
1.95	3.96875	2.98625
1.99562	3.99609	2.99903
1.99927	3.99951	2.9998
1.99993	3.99994	2.99998
1.99999	3.99999	3
2	4	3
2	4	3
2	4	3
2	4	3
2	4	3
2	4	3

כאשר כל שורה מייצגת את ערכי הקואורדינטות של הוקטור x באותה אירטציה.

ניתן לראות שלאחר 13 איטרציות שתי השיטות התכנסו לאותו הפתרון : $x = \begin{pmatrix} 2 \\ 4 \\ 3 \end{pmatrix}$, אשר הינו פתרון נכון. כמו כן, שיטת $Gauss - Siedel$ התכנסה לפתרון פי שתיים מהר יותר מאשר שיטת $Jacobi$.

שאלה 5:

בהינתן מערכת המשוואות הבאה:

$$\begin{cases} 1.133x_1 + 5.281x_2 = 6.414 \\ 24.14x_1 - 1.210x_2 = 22.93 \end{cases}$$

נפתור אותה בשיטת $Gaussian\ Elimination$ בדיוק של 4 ספרות עשרוניות:

- $without\ pivoting$:

$Row\ Operations$	$Augmented\ Matrix$
	$\begin{bmatrix} 1.133 & 5.281 & 6.414 \\ 24.14 & -1.210 & 22.93 \end{bmatrix}$
$L_2 - \frac{24.14}{1.133}L_1 \to L_2$	$\begin{bmatrix} 1.133 & 5.281 & 6.414 \\ 0 & -113.7 & -113.8 \end{bmatrix}$

$$\boldsymbol{x_2} = \frac{-113.8}{-113.7} = \mathbf{1.001}$$

$$\boldsymbol{x_1} = \frac{6.414 - 5.281x_2}{1.133} = \mathbf{0.9956}$$

- $with\ partial\ pivoting$:

$Row\ Operations$	$Augmented\ Matrix$
	$\begin{bmatrix} 1.133 & 5.281 & 6.414 \\ 24.14 & -1.210 & 22.93 \end{bmatrix}$
$L_1 \leftrightarrow L_2$	$\begin{bmatrix} 24.14 & -1.210 & 22.93 \\ 1.133 & 5.281 & 6.414 \end{bmatrix}$
$L_2 - \frac{1.133}{24.14}L_1 \to L_2$	$\begin{bmatrix} 24.14 & -1.210 & 22.93 \\ 0 & 5.338 & 5.338 \end{bmatrix}$

$$\boldsymbol{x_2} = \frac{5.338}{5.338} = \mathbf{1.000}$$

$$\boldsymbol{x_1} = \frac{22.93 - (-1.210)x_2}{24.14} = \mathbf{1.000}$$

שאלה 6:

בהינתן המטריצה $\begin{pmatrix} 5 & -1 & 7 \\ -1 & -1 & 1 \\ 7 & 1 & 5 \end{pmatrix}$, אנו מתבקשים להשתמש ב- $Power\ method$ עם 9 איטרציות כדי למצוא עבורה קירוב לערך עצמי ווקטור עצמי.

להלן תוכנית פיתון המבצעת את החישוב:

```python
import numpy as np
from tabulate import tabulate


def print_tables(L, V):
    np.set_printoptions(linewidth=np.inf)
    print(f'L = {L}')
    headers = ['x1', 'x2', 'x3']
    print(f'V =\n{tabulate(V, headers, tablefmt="fancy_grid")}')


def main():
    iter = 9
    A = np.array([[5, -1, 7], [-1, -1, 1], [7, 1, 5]])
    x = np.array([1, 1, 1])  # initial guess
    L = np.zeros(iter)
    V = np.zeros((iter, 3))

    for i in range(0, iter):
        y = np.matmul(A, x)
        L[i] = max(np.absolute(np.amax(y)), np.absolute(np.amin(y)))
        x = np.divide(y, L[i])  # scaling
        V[i] = x

    print_tables(L, V)


if __name__ == "__main__":
    main()
```

פלט התוכנית הינו:

```
L = [13.         11.30769231 11.81632653 11.95336788 11.98829649 11.99707127 11.99926764 11.9998169  11.99995422]
V =
```

x1	x2	x3
0.846154	-0.0769231	1
1	0.0204082	0.959184
0.989637	-0.00518135	1
1	0.00130039	0.997399
0.999349	-0.000325415	1
1	8.13736e-05	0.999837
0.999959	-2.03446e-05	1
1	5.08624e-06	0.99999
0.999997	-1.27156e-06	1

הרשימה L מחזיקה את ערכי הערך העצמי והרשימה V מחזיקה את ערכי הוקטור העצמי, לאורך האיטרציות.

הערכים שקיבלנו באיטרציה האחרונה הינם : ערך עצמי 11.99995, ווקטור עצמי $\begin{pmatrix} 0.99999 \\ 0.00000 \\ 1 \end{pmatrix}$.

הערכים המקורבים שהפונקציה מצאה קרובים מאוד לערכים הנכונים : ערך עצמי 12 עם וקטור עצמי $\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$.