МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Інститут **КНІТ**
Кафедра **ПЗ**

ЗВІТ

До лабораторної роботи № 5
На тему: “*Наближені методи розв'язування систем лінійних алгебраїчних рівнянь*”
З дисципліни: “Чисельні методи”

Лектор:
доцент кафедри ПЗ
Мельник Н.Б.

Виконав:
студент групи ПЗ-16
Коваленко Д.М.

Прийняв:
асистент кафедри ПЗ
Гарматій Г.Ю.

«_____» ________ 2022 р.
$\sum$ = _____................

Львів — 2022

Тема. Наближені методи розв'язування систем лінійних алгебраїчних рівнянь.

Мета. Ознайомлення на практиці з методами Якобі та Зейделя розв'язування систем лінійних алгебраїчних рівнянь.

Теоретичні відомості

Метод Якобі

Розглянемо систему лінійних алгебраїчних рівнянь:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \alpha_{13}x_3 + ... + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \alpha_{23}x_3 + ... + a_{2n}x_n = b_2, \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ a_{n1}x_1 + a_{n2}x_2 + \alpha_{n3}x_3 + ... + a_{nn}x_n = b_n. \end{cases}$$

Припустивши, що коефіцієнти $a_{ii} \neq 0 \quad (i = \overline{1,n})$, розв'яжемо i-те рівняння системи відносної x_i та введемо позначення:

$$\beta_i = \frac{b_i}{a_{ii}}, \quad a_{ij} = -\frac{a_{ij}}{a_{ii}}, \quad i = \overline{1,n}, \quad j \neq i$$

Отримаємо:

$$\begin{cases} x_1 = \beta_1 + \alpha_{12}x_2 + \alpha_{13}x_3 + ... + \alpha_{1n}x_n, \\ x_2 = \beta_2 + \alpha_{21}x_1 + \alpha_{23}x_3 + ... + a_{2n}x_n, \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ x_n = \beta_n + \alpha_{n2}x_2 + \alpha_{n3}x_3 + ... + a_{nn}x_n. \end{cases}$$

Ітераційні формули мають вигляд:

$$\begin{cases} x_i^{(0)} = \beta_i, \\ x_i^{(k)} = \beta_i + \sum_{j=1}^{i-1} a_{ij}x_j^{(k-1)} + \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)}, \quad i = \overline{1,n}, \quad k = 1, 2... \end{cases}$$

Метод Зейделя

Основна відмінність методу Зейделя від методу Якобі полягає в тому, що для обчислення чергового наближення розв'язку використовуються вже знайдені значення.

Ітераційні формули мають вигляд:

$$\begin{cases} x_i^{(0)} = \beta_i, \\ x_i^{(k)} = \beta_i + \sum_{j=1}^{i-1} a_{ij}x_j^{(k-1)} + \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)}, \quad i = \overline{1,n}, \quad k = 1, 2... \end{cases}$$

Збіжність ітераційного процесу

Якщо елементи матриці α системи рівнянь задовольняють одну з умов:

$$\sum_{j=1}^{n} |\frac{a_{ij}}{a_{ii}}| < 1, \quad i = \overline{1,n}$$

$$\sum_{i=1}^{n} |\frac{a_{ij}}{a_{ii}}| < 1, \quad j = \overline{1,n}$$

$$\sum_{i,j=1}^{n} (\frac{a_{ij}}{a_{ii}})^2 < 1.$$

то система рівнянь має єдиний розв'язок X^*, який не залежить від початкового наближення $X^{(0)}$.

Критерії припинення ітераційного процесу

Якщо задана похибка ϵ наближеного розв'язку, то критерієм припинення ітераційного процесу вважають виконання однієї з умов:

$$|X^{(k)} - X^{(k-1)}| = \sqrt{\sum_{i=1}^{n}(x_i^{(k)} - x_i^{(k-1)})^2} < \epsilon, \quad k = 1, 2...$$

$$max|x_i^{(k)} - x_i^{(k-1)}| < \epsilon, \quad i = \overline{1,n}, \quad k = 1, 2...$$

$$max|\frac{x_i^{(k)} - x_i^{(k-1)}}{x_i^{(k)}}| < \epsilon, \quad |x_i| >> 1, \quad i = \overline{1,n}, \quad k = 1, 2...$$

Лабораторне завдання

Розв'язати систему лінійних алгебраїчних рівнянь методом Зейделя та методом Якобі з точністю $\epsilon = 0.001$. Порівняти кількість ітерацій для обох методів.

$$\begin{cases} 0.65x_1 - 0.06x_2 - 0.12x_3 + 0.14x_4 = 2.17 \\ 0.04x_1 - 0.82x_2 + 0.08x_3 + 0.11x_4 = -1.14 \\ 0.34x_1 + 0.08x_2 - 0.66x_3 + 0.14x_4 = 2.1 \\ 0.11x_1 + 0.12x_2 \qquad\qquad - 0.53x_4 = -0.58 \end{cases}$$

Хід роботи

Метод Якобі

$$\begin{cases} 0.65x_1 - 0.06x_2 - 0.12x_3 + 0.14x_4 = 2.17 \\ 0.04x_1 - 0.82x_2 + 0.08x_3 + 0.11x_4 = -1.14 \\ 0.34x_1 + 0.08x_2 - 0.66x_3 + 0.14x_4 = 2.1 \\ 0.11x_1 + 0.12x_2 \qquad\qquad - 0.53x_4 = -0.58 \end{cases}$$

Перепишу систему у зведеному вигляді:

$$\begin{cases} x_1 = 3.33 + 0.09x_2 + 0.18x_3 - 0.21x_4 \\ x_2 = 1.39 + 0.04x_1 + 0.09x_3 + 0.13x_4 \\ x_3 = -3.18 + 0.51x_1 + 0.12x_2 + 0.21x_4 \\ x_4 = 1.09 + 0.20x_1 + 0.22x_2 \end{cases}$$

Звідси:

$$\alpha = \begin{pmatrix} 0 & 0.09 & 0.18 & -0.21 \\ 0.04 & 0 & 0.09 & 0.13 \\ 0.51 & 0.12 & 0 & 0.21 \\ 0.20 & 0.22 & 0 & 0 \end{pmatrix} \qquad \beta = \begin{pmatrix} 3.33 \\ 1.39 \\ -3.18 \\ 1.09 \end{pmatrix}$$

Достатня умова збіжності методу простої ітерації виконується:

$$\max\{|0.09| + |0.18| + |-0.21|, |0.04| + |0.09| + |0.13|, |0.51| + |0.12| + |0.21|, |0.20| + |0.22|\} =$$
$$= \max\{0.48, 0.26, 0.82, 0.42\} = 0.82 < 1$$

За початкове наближення візьму стовпець вільних членів β:

$$\begin{cases} x_1 = 3.33 \\ x_2 = 1.39 \\ x_3 = -3.18 \\ x_4 = 1.09 \end{cases}$$

Перший крок ітераційного процесу:

$$\begin{cases} x_1 = 3.33 + 0.09 \cdot 1.39 + 0.18 \cdot (-3.18) - 0.21 \cdot 1.09 \\ x_2 = 1.39 + 0.04 \cdot 3.33 + 0.09 \cdot (-3.18) + 0.13 \cdot 1.09 \\ x_3 = -3.18 + 0.51 \cdot 3.33 + 0.12 \cdot 1.39 + 0.21 \cdot 1.09 \\ x_4 = 1.09 + 0.20 \cdot 3.33 + 0.22 \cdot 1.39 \end{cases} \qquad \begin{cases} x_1 = 2.64 \\ x_2 = 1.38 \\ x_3 = -1.06 \\ x_4 = 2.10 \end{cases}$$

Другий крок ітераційного процесу:

$$\begin{cases} x_1 = 3.33 + 0.09 \cdot 1.38 + 0.18 \cdot (-1.06) - 0.21 \cdot 2.10 \\ x_2 = 1.39 + 0.04 \cdot 2.64 + 0.09 \cdot (-1.06) + 0.13 \cdot 2.10 \\ x_3 = -3.18 + 0.51 \cdot 2.64 + 0.12 \cdot 1.38 + 0.21 \cdot 2.10 \\ x_4 = 1.09 + 0.20 \cdot 2.64 + 0.22 \cdot 1.38 \end{cases} \qquad \begin{cases} x_1 = 2.81 \\ x_2 = 1.69 \\ x_3 = -1.20 \\ x_4 = 1.95 \end{cases}$$

...

Результат після кількох додаткових ітерацій:

$$\begin{cases} x_1 = 2.85 \\ x_2 = 1.70 \\ x_3 = -1.06 \\ x_4 = 2.07 \end{cases}$$

Код програми *(файл lab_51.py):*

```
import numpy as np

def data_to_matrix(path):
    return (
        np.loadtxt(open(path,"rb"),delimiter=",",usecols=[0,1,2,3]),
        np.loadtxt(open(path,"rb"),delimiter=",",usecols=4),
    )

def jacobi_method(A, B, eps):
    """ Метод Якобі """
    print(jacobi_method.__doc__)
    D = np.diag(A)
    a = np.array([-(A - np.diagflat(D))[i] / D[i] for i in range(np.
  shape(A)[0])])      # α
    b = B / D            # β

    print(f"α:\n {a}\n β:\n {b}")

    X = b.copy() # Початкове наближення

    for i in range(10):
        print(f"\Ітераціяt {i}: {X}")
        X_new = b + np.dot(a, X)
        if np.allclose(X_new, X, atol=eps):
            break
        X = X_new.copy()
    print(f"Відповідь: {X}")

path = input("Введіть шлях до файлу з даними: ") or "data.csv"
A, B = data_to_matrix(path)       # A - матриця коефіцієнтів системи
jacobi_method(A, B, 0.001)        # B - матриця вільних членів
```

Метод Зейделя

Основна відмінність методу Зейделя від методу Якобі полягає в тому, що для обчислення чергового наближення розв'язку використовуються вже знайдені значення.

Перший крок ітераційного процесу:

$$\begin{cases} x_1 = 3.33 + 0.09 \cdot 1.39 + 0.18 \cdot (-3.18) - 0.21 \cdot 1.09 = \mathbf{2.64} \\ x_2 = 1.39 + 0.04 \cdot \mathbf{2.64} + 0.09 \cdot (-3.18) + 0.13 \cdot 1.09 = \mathbf{1.38} \\ x_3 = -3.18 + 0.51 \cdot \mathbf{2.64} + 0.12 \cdot \mathbf{1.38} + 0.21 \cdot 1.09 = \mathbf{-1.06} \\ x_4 = 1.09 + 0.20 \cdot \mathbf{2.64} + 0.22 \cdot \mathbf{1.38} = \mathbf{2.10} \end{cases}$$

Другий крок ітераційного процесу:

$$\begin{cases} x_1 = 3.33 + 0.09 \cdot 1.38 + 0.18 \cdot (-1.06) - 0.21 \cdot 2.10 = \mathbf{2.81} \\ x_2 = 1.39 + 0.04 \cdot \mathbf{2.81} + 0.09 \cdot (-1.06) + 0.13 \cdot 2.10 = \mathbf{1.69} \\ x_3 = -3.18 + 0.51 \cdot \mathbf{2.81} + 0.12 \cdot \mathbf{1.69} + 0.21 \cdot 2.10 = \mathbf{-1.20} \\ x_4 = 1.09 + 0.20 \cdot \mathbf{2.81} + 0.22 \cdot \mathbf{1.69} = \mathbf{1.95} \end{cases}$$

...

Код програми *(файл lab_52.py):*

```
import numpy as np

def data_to_matrix(path):
    return (
        np.loadtxt(open(path,"rb"),delimiter=",",usecols=[0,1,2,3]),
        np.loadtxt(open(path,"rb"),delimiter=",",usecols=4),
    )

def seidel_method(A, B, eps):
    """ Метод Зейделя """
    print(seidel_method.__doc__)
    X = B / np.diag(A) # Початкове наближення

    for i in range(1000):
        print(f"\Ітераціяt {i}: {X}")
        X_prev = X.copy()
        for k in range(A.shape[0]):
            X[k] = (B[k] - np.dot(A[k, :k], X[:k]) - np.dot(A[k, k
    +1:], X_prev[k+1:])) / A[k, k]
            if np.allclose(X, X_prev, atol=eps):
                break
    print(f"Відповідь: {X}")

path = input("Введіть шлях до файлу з даними: ") or "data.csv"
A, B = data_to_matrix(path)        # A - матриця коефіцієнтів системи
seidel_method(A, B, 0.001)         # B - матриця вільних членів
```

```
Введіть шлях до файлу з даними: data.csv
 Метод Якобі
α:
[[-0.          0.09230769  0.18461538 -0.21538462]
 [ 0.04878049  0.          0.09756098  0.13414634]
 [ 0.51515152  0.12121212  0.          0.21212121]
 [ 0.20754717  0.22641509  0.          0.        ]]
β:
[ 3.33846154  1.3902439  -3.18181818  1.09433962]
        Ітерація 0: [ 3.33846154  1.3902439  -3.18181818  1.09433962]
        Ітерація 1: [ 2.64367524  1.38947606 -1.0613576   2.10200007]
        Ітерація 2: [ 2.81803945  1.69763221 -1.20562473  1.95762529]
        Ітерація 3: [ 2.85094681  1.67269559 -1.10907344  2.06358529]
        Ітерація 4: [ 2.84364766  1.69793461 -1.07266742  2.06476909]
        Ітерація 5: [ 2.85244356  1.70128916 -1.0731172   2.06896867]
        Ітерація 6: [ 2.85176565  1.70223771 -1.06728855  2.07155376]
Відповідь: [ 2.85176565  1.70223771 -1.06728855  2.07155376]
```

(а)

```
Введіть шлях до файлу з даними: data.csv
 Метод Зейделя
        Ітерація 0: [ 3.33846154  1.3902439  -3.18181818  1.09433962]
        Ітерація 1: [ 2.64367524  1.35558404 -1.42347901  1.94995162]
        Ітерація 2: [ 2.78080667  1.64859588 -1.13582551  2.04475517]
        Ітерація 3: [ 2.84053995  1.69229101 -1.07964759  2.06704588]
        Ітерація 4: [ 2.85014358  1.70123047 -1.06888837  2.07106311]
        Ітерація 5: [ 2.85208983  1.70291398 -1.06682955  2.07184822]
Відповідь: [ 2.85208983  1.70291398 -1.06682955  2.07184822]
```

(б)

```
0.65, -0.06, -0.12, 0.14, 2.17
0.04, -0.82, 0.08, 0.11, -1.14
0.34, 0.08, -0.66, 0.14, 2.1
0.11, 0.12, 0.0, -0.53, -0.58
```

(в)

Рис. 1: Метод Якобі (а), метод Зейделя (б), файл даних (в)

Висновок

На лабораторній роботі я засвоїв практичні навички використання методу Якобі та методу Зейделя та розробив функції для розв'язку системи лінійних алгебраїчних рівнянь

$$\begin{cases} 0.65x_1 - 0.06x_2 - 0.12x_3 + 0.14x_4 = 2.17 \\ 0.04x_1 - 0.82x_2 + 0.08x_3 + 0.11x_4 = -1.14 \\ 0.34x_1 + 0.08x_2 - 0.66x_3 + 0.14x_4 = 2.1 \\ 0.11x_1 + 0.12x_2 \qquad\qquad\;\; - 0.53x_4 = -0.58 \end{cases}$$

за допомогою цих методів з точністю $\epsilon = 0.001$. Корені системи рівнянь: $2.85; 1.70; -1.06; 2.07$. В результаті виконання програми видно, що кількість ітерацій методу Зейделя менша ніж методу Якобі.