Національний університет «Львівська політехніка»
Інститут комп’ютерних наук та інформаційних технологій

ЗВІТ
Про виконання лабораторної роботи № 7
«Чисельні методи розв’язування систем нелінійних рівнянь»
з дисципліни «Чисельні методи»

Лектор:
доцент кафедри ПЗ
Мельник Н.Б.

Виконав:
студ. групи ПЗ-15
Марущак А. С.

Прийняв:
асистент кафедри ПЗ
Гарматій Г.Ю
«___» ______ 2022 р.
∑ = ____ ___________

Львів – 2022

Тема роботи: Чисельні методи розв’язування систем нелінійних рівнянь.

Мета роботи: ознайомлення на практиці з методом ітерацій та методом Ньютона розв’язування систем нелінійних рівнянь.

Теоретичні відомості

Розглянемо систему двох нелінійних рівнянь з двома невідомими

$$\begin{cases} f_1(x,y)=0, \\ f_2(x,y)=0. \end{cases}$$

Розв’язком цієї системи є пара чисел (x_*, y_*), яка перетворює систему рівнянь в тотожність.

Метод ітерації

Припустимо, що (x_0, y_0) - наближений розв’язок системи, яку перетворимо до такого вигляду

$$\begin{cases} x=\varphi_1(x,y), \\ y=\varphi_2(x,y), \end{cases}$$

де φ_1 , φ_2 - неперервно-диференційовані функції за змінними x та y .
Розглянемо ітераційний процес

$$\begin{cases} x_{n+1}=\varphi_1(x_n,y_n), \\ y_{n+1}=\varphi_2(x_n,y_n), \end{cases} \qquad n=1,\,2,\dots$$

який породжує числові послідовності $\{x_n\}$, $\{y_n\}$.

Якщо ітераційний процес збігається, тобто існують границі

$$x_*=\lim_{n\to\infty}x_n\,, \qquad y_*=\lim_{n\to\infty}y_n\,,$$

То розв’язком початкової системи є пара чисел (x_*, y_*).

Причому, умовами збіжності цього процесу в деякій замкнутій області D $\{a \le x \le A, b \le y \le b\}$ є наступні:

1) функції $\varphi_1(x,y), \varphi_2(x.y)$ визначені та неперервно-диференційовані в області D;
2) початкове наближення (x_0, y_0) і всі наступні наближення (x_n, y_n) (n =1, 2,...) належать області D;
3) в області D виконуються нерівності

$$\left|\frac{\partial\varphi_1}{\partial x}\right| + \left|\frac{\partial\varphi_2}{\partial x}\right| \le q_1 < 1, \qquad \left|\frac{\partial\varphi_1}{\partial y}\right| + \left|\frac{\partial\varphi_2}{\partial y}\right| \le q_2 < 1$$

Або аналогічні

$$\left|\frac{\partial\varphi_1}{\partial x}\right| + \left|\frac{\partial\varphi_1}{\partial y}\right| \le q_1^* < 1, \qquad \left|\frac{\partial\varphi_2}{\partial x}\right| + \left|\frac{\partial\varphi_2}{\partial y}\right| \le q_2^* < 1$$

Метод Ньютона

Це найрозповсюдженіший метод розв'язування систем нелінійних рівнянь. Він забезпечує кращу збіжність, ніж метод простої ітерації.

Нехай (x_0, y_0) - наближений розв'язок системи, а Δx Δy , - деякі поправки до точного розв'язку.

Розглянемо систему рівнянь

$$\begin{cases} f_1(x+\Delta_x, y+\Delta_y)=0, \\ f_2(x+\Delta_x, y+\Delta_y)=0. \end{cases}$$

Розкладемо функції $f_1,\ f_2$ в ряд Тейлора, обмежившись лінійними членами розкладу відносно Δx, Δy

$$\begin{cases} f_1(x_0, y_0)+\dfrac{\partial f_1}{\partial x}\Delta_x+\dfrac{\partial f_1}{\partial y}\Delta_y=0, \\ f_2(x_0, y_0)+\dfrac{\partial f_2}{\partial x}\Delta_x+\dfrac{\partial f_2}{\partial y}\Delta_y=0. \end{cases}$$

І тоді цю систему перепишемо у вигляді

$$\begin{cases} \dfrac{\partial f_1}{\partial x}\Delta_x+\dfrac{\partial f_1}{\partial y}\Delta_y=-f_1(x_0, y_0), \\ \dfrac{\partial f_2}{\partial x}\Delta_x+\dfrac{\partial f_2}{\partial y}\Delta_y=-f_2(x_0, y_0). \end{cases}$$

Тоді, розв'яжемо цю систему відносно Δ_x, Δ_y будь-яким відномим нам методом розв'язування СЛАР.

І тоді наступне наближення виражається через попереднє наступним чином

$$x_{n+1}=x_n+\Delta_x \qquad y_{n+1}=y_n+\Delta_y \qquad n=0,1,2,\ldots$$

Ці операції треба повторювати, допоки не досягнемо потрібної точності.

Індивідуальне завдання

1. Ознайомитися з теоретичним матеріалом.
2. Розв'язати систему нелінійних рівнянь з точністю $\varepsilon = 10^{-3}$ методом ітерацій та методом Ньютона.

Варіант завдання

$$\begin{cases} \cos(x+0{,}5)+y=0{,}8 \\ \sin y - 2x = 1{,}6 \end{cases}$$

Хід роботи

Проведемо деякі обчилення вручну, щоб потім мати змогу використати отримані результати в реалізації програми.

По перше, нам необхідно локалізувати розв'язок системи. Для цього скористаємось графічним методом. Побудуємо графіки цих 2-ох рівнянь на координатній площині xOy:

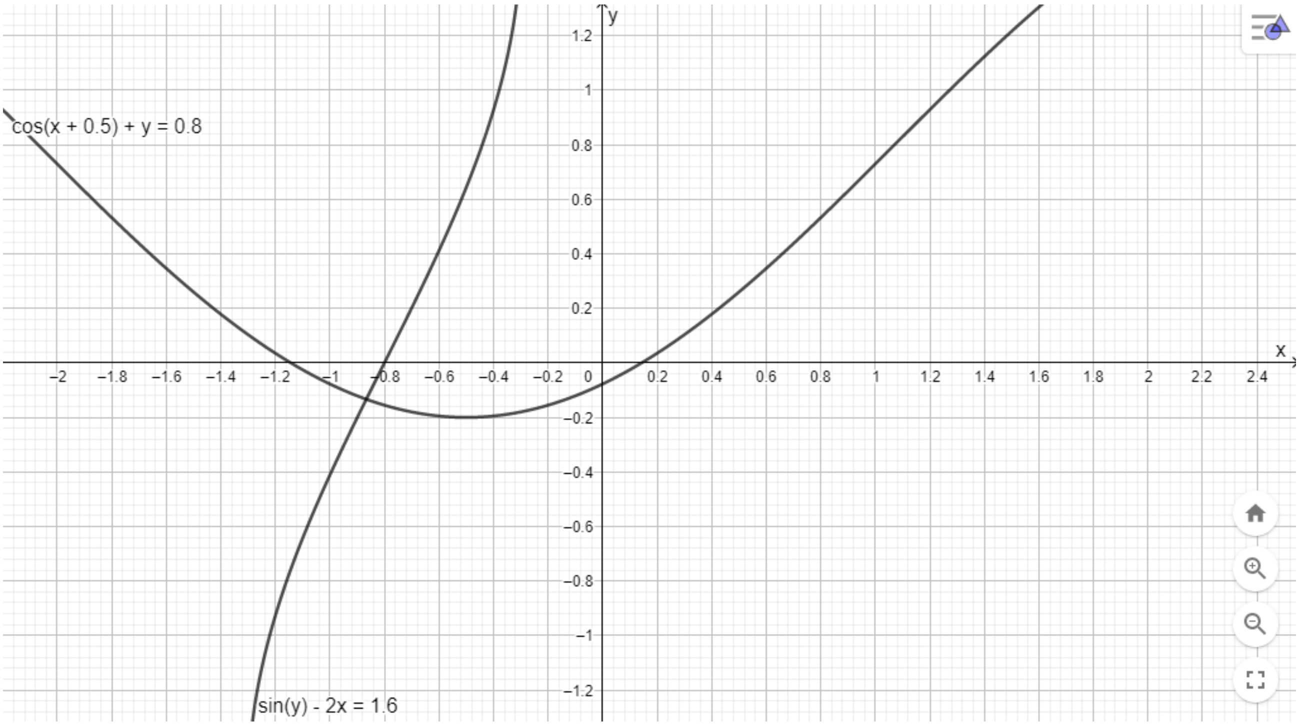

Рис 1.1 Графічна інтерпретація системи рівнянь

З рис 1.1 очевидно, що корінь локалізований в області $D\ \{-1 \le x \le 1;\ -1 \le y \le 1\}$, а також видно, що за початкове наближення можна взяти точку $(x_0, y_0) = (-0.8, -0.1)$.

Тепер для того, щоб розв'язати систему методом ітерації, запишемо її у наступному вигляді:

$$\begin{cases} y = 0.8 - \cos(x+0.5) \\ x = \dfrac{1}{2}\sin y - 0.8 \end{cases}$$

Введемо позначення

$$\varphi_1(x,y) = \frac{1}{2}\sin y - 0.8, \quad \varphi_2(x,y) = 0.8 - \cos(x+0.5)$$

і, для зручності, переставимо рядки системи місцями:

$$\begin{cases} x = \varphi_1(x,y) \\ y = \varphi_2(x,y) \end{cases}.$$

Перевіримо збіжність системи на області D:

$$\left|\frac{\partial\varphi_1}{\partial x}\right| + \left|\frac{\partial\varphi_1}{\partial y}\right| = 0 + \frac{1}{2}\cos y \le \frac{1}{2}\cos 0 = 0.5 < 1$$

$$\left|\frac{\partial\varphi_2}{\partial x}\right| + \left|\frac{\partial\varphi_2}{\partial y}\right| = \sin(x + 0{,}5) + 0 \le \sin(1{,}5) < \sin\left(\frac{\pi}{2}\right) = 1.$$

Отож, ітераційний процес

$$\begin{cases} x_{n+1} = \varphi_1(x,y) \\ y_{n+1} = \varphi_2(x,y) \end{cases}, n = 0,1,2,\ldots$$

буде збіжним на області $D\,\{-1 \le x \le 1;\ -1 \le y \le 1\}$ при початковому наближені $(x_0, y_0) = (-0.8, -0.1)$. Для прикладу, проведемо першу ітерацію:

$$x_1 = \frac{1}{2}\sin(-0.1) - 0.8 \approx -0.85; \quad y_1 = 0.8 - \cos(-0.8 + 0.5) = -0.16$$

Подальші ітерації виконаємо вже програмно.

Для методу Ньютона перепишемо вихідну систему у вигляді:

$$\begin{cases} \cos(x + 0.5) + y - 0.8 = 0 \\ \sin(y) - 2x - 1.6 = 0 \end{cases}$$

Тоді введемо позначення $f_1(x,y) = \cos(x + 0.5) + y - 0.8, f_2(x,y) = \sin(y) - 2x - 1.6$. Отримаємо, що

$$\begin{cases} f_1(x,y) = 0 \\ f_2(x,y) = 0 \end{cases}$$

Тоді, нехай числа Δ_x, Δ_y такі, що

$$\begin{cases} f_1(x_0 + \Delta_x, y_0 + \Delta_y) = 0 \\ f_2(x_0 + \Delta_x, y_0 + \Delta_y) = 0 \end{cases}$$

Тоді, за розкладом Тейлора, обмежившись лінійними членами, цю систему можна записати так:

$$\begin{cases} f_1(x_0, y_0) + \dfrac{\partial f_1}{\partial x}\Delta_x + \dfrac{\partial f_1}{\partial y}\Delta_y = 0, \\ f_2(x_0, y_0) + \dfrac{\partial f_2}{\partial x}\Delta_x + \dfrac{\partial f_2}{\partial y}\Delta_y = 0. \end{cases}$$

Або у вигляді

$$\begin{cases} -\sin(x_0 + 0.5)\,\Delta_x + \Delta_y = -\cos(x_0 + 0.5) - y_0 + 0.8, \\ -2\Delta_x + \cos(y_0)\,\Delta_y = -\sin(y_0) + 2x_0 + 1.6. \end{cases}$$

Маємо СЛАР відносно Δ_x, Δ_y. Для n-ої ітерації $(n = 0,1,2,3,\ldots)$ маємо:

$$\begin{cases} -\sin(x_n + 0.5)\,\Delta_x + \Delta_y = -\cos(x_n + 0.5) - y_n + 0.8, \\ \quad -2\Delta_x + \cos(y_n)\,\Delta_y = -\sin(y_n) + 2x_n + 1.6, \end{cases}$$

Де $x_n = x_{n-1} + \Delta_x; y_n = y_{n-1} + \Delta_y$. Або у матричному вигляді:

$$J(x_n, y_n) \cdot P = -F(x_n, y_n),$$

Де $J(x_n, y_n) = \begin{pmatrix} -\sin(x_n + 0.5) & 1 \\ -2 & \cos(y_n) \end{pmatrix}$ – матриця коефіцієнтів, або так звана матриця Якобі, $P = \begin{pmatrix} \Delta_x \\ \Delta_y \end{pmatrix}$, $F(x_n, y_n) = \begin{pmatrix} \cos(x + 0.5) + y - 0.8 \\ \sin(y) - 2x - 1.6 \end{pmatrix}$. Тоді, згідно з методом оберненої матриці запишемо, що

$$P = -J(x_n, y_n)^{-1} \cdot F(x_n, y_n).$$

І, врахувавши формулу переходу від x_{n-1} до x_n, y_{n-1} до y_n, запишемо ітераційну формулу:

$$X^{(n)} = X^{(n-1)} - J(x_{n-1}, y_{n-1})^{-1} \cdot F(x_{n-1}, y_{n-1}), n = 1,2,3, \dots,$$

де $X^{(n)} = \begin{pmatrix} x_n \\ y_n \end{pmatrix}$.

Наприклад, для першої ітерації $J(x_0, y_0) = \begin{pmatrix} 0.296 & 1 \\ -2 & 0.995 \end{pmatrix} \to J(x_0, y_0)^{-1} = \begin{pmatrix} 0.434 & -0.436 \\ 0.872 & 0.129 \end{pmatrix}$, $F(x_0, y_0) = \begin{pmatrix} 0.055 \\ -0.1 \end{pmatrix}$. Тоді

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} -0.8 \\ -0.1 \end{pmatrix} - \begin{pmatrix} 0.296 & 1 \\ -2 & 0.995 \end{pmatrix} \begin{pmatrix} 0.055 \\ -0.1 \end{pmatrix} = \begin{pmatrix} -0.868 \\ -0.135 \end{pmatrix}$$

Наступна мета – реалізувати подані методи програмно, використовуючи ці дані.

Для подальшої роботи я буду використовувати структуру даних Matrix, яка відповідає матриці заданого розміру, як зрозуміло з назви. Серед її функціональних можливостей – множення та додавання матриць, пошук транспонованої матриці та детермінанту, пошук оберненої матриці. Покажемо це у коді:

```
public static Matrix operator*(Matrix a, Matrix b)
        {
            if (a.ColumnsCount != b.RowsCount)
                throw new ArgumentException("Cannot multiply matrices of these sizes.");
            Matrix result = new(a.RowsCount, b.ColumnsCount);

            for (int i = 0; i < result.RowsCount; i++)
                for (int j = 0; j < result.ColumnsCount; j++)
                    result[i, j] = a.Row(i) * b.Column(j);

            return result;
        }

public static Matrix operator+(Matrix a, Matrix b)
        {
            if (a.ColumnsCount != b.ColumnsCount || a.RowsCount != b.RowsCount)
                throw new ArgumentException("Cannot add matrices of different sizes.");

            Matrix result = new(a.RowsCount, b.ColumnsCount);
```

```
            for (int i = 0; i < result.RowsCount; i++)
                for (int j = 0; j < result.ColumnsCount; j++)
                    result[i, j] = a[i, j] + b[i, j];

            return result;
        }


public Matrix Transponed()
        {
            Matrix result = new(ColumnsCount, RowsCount);

            for (int i = 0; i < RowsCount; i++)
                for (int j = 0; j < ColumnsCount; j++)
                    result[j, i] = this[i, j];

            return result;
        }

public double Determinant()
        {
            if (ColumnsCount != RowsCount)
                throw new InvalidOperationException("Cannot find determinant for non-square
matrices.");
            if (ColumnsCount == 1)
                return this[0, 0];

            double result = 0;

            for (int i = 0; i < ColumnsCount; i++)
            {
                result += Math.Pow(-1, i) * this[0, i] *
this.WithoutRow(0).WithoutColumn(i).Determinant();
            }

            return result;
        }

public double AlgebraicAddition(int i, int j) => Math.Pow(-1, i+j) *
this.WithoutRow(i).WithoutColumn(j).Determinant();

public Matrix Inversed()
        {
            if (RowsCount != ColumnsCount)
                throw new InvalidOperationException();
            Matrix result = new Matrix(RowsCount, ColumnsCount);

            for(int i = 0; i < RowsCount; i++)
            {
                for(int j = 0; j < ColumnsCount; j++)
                {
                    result[i, j] = this.AlgebraicAddition(i, j);
                }
            }

            return (1/this.Determinant()) * (result.Transponed());
        }
```

І основна програма буде виглядати наступним чином:

```
using MatrixLib;

Console.Write("Початкове наближення: ");
```

```
var numbers = Console.ReadLine().Split().Select(double.Parse).Take(2).ToArray();

var x0 = numbers[0];
var y0 = numbers[1];

var solution = SimpleIteration(x0, y0, out int iters);
Console.WriteLine($"Метод ітерації: {solution.Item1:0.000}, {solution.Item2:0.000};
Ітерацій: {iters}");
var solution2 = NewtonMethod(x0, y0, out int iters2);
Console.WriteLine($"Метод Ньютона: {solution2.Item1:0.000}, {solution2.Item2:0.000};
Ітерацій: {iters2}");

(double, double) SimpleIteration(double x0, double y0, out int iterations)
{
    double x = x0, y = y0;
    iterations = 0;
    do
    {
        //Зберігаємо попередні значення
        x0 = x;
        y0 = y;
        //Використовуємо ітераційні формули
        x = Math.Sin(y0) / 2 - 0.8;
        y = 0.8 - Math.Cos(x0 + 0.5);
        iterations++;
    } while (Math.Max(Math.Abs(y - y0),Math.Abs(x - x0)) > 0.001); // Допоки не досягнемо
заданої точності.

    return (x, y);
}

//Обчислення матриці Якобі
Matrix J(Matrix X)
{
    Matrix res = new(2, 2);
    res[0, 0] = -Math.Sin(X[0, 0] + 0.5);
    res[0, 1] = 1;
    res[1, 0] = -2;
    res[1, 1] = Math.Cos(X[1, 0]);

    return res;
}

//Обчислення значень функції
Matrix F(Matrix X)
{
    Matrix res = new(2, 1);
    res[0, 0] = Math.Cos(X[0, 0] + 0.5) + X[1, 0] - 0.8;
    res[1, 0] = Math.Sin(X[1, 0]) - 2 * X[0, 0] - 1.6;
    return res;
}

(double, double) NewtonMethod(double x0, double y0, out int iterations)
{
    Matrix X = new(2, 1), X_prev;
    X[0,0] = x0;
    X[1, 0] = y0;

    iterations = 0;
    do
    {
        //Зберігаємо попередні значення
        X_prev = X.Clone();
        //Використовуємо ітераційну формулу
        X = X_prev + (-1) * J(X_prev).Inversed() * F(X_prev);
```

```
        iterations++;
    } while (Math.Max(Math.Abs(X_prev[0, 0] - X[0, 0]), Math.Abs(X_prev[1, 0] - X[1, 0])) >
0.001); // Допоки не досягнемо заданої точності.

    return (X[0,0], X[1,0]);
}
```

Як зазначалось вище, початкове наближення $(x_0, y_0) = (-0.8, -0.1)$. Подамо його на вхід програми і подивимось на результат.

<u>Результат</u> виконання програми:

```
Початкове наближення: -0,8 -0,1
Метод ітерації: -0,867, -0,134; Ітерацій: 6
Метод Ньютона: -0,867, -0,134; Ітерацій: 3
```

Рис 1.2 Результат виконання програми.

<u>Аналіз результатів</u>:

Як бачимо, результати обчислень обома методами співпали. Для того щоб перевірити їх дійсність, підставимо їх у початкову систему:

$$\begin{cases} \cos(-0{,}867 + 0{,}5) - 0{,}134 \approx 0{,}8 \\ \sin(-0{,}134) - 2 \cdot (-0{,}867) \approx 1{,}6 \end{cases}.$$

Це означає, що систему було розв’язано вірно.

<u>Висновок</u>:

Виконуючи цю лабораторну роботу ми ознайомились на практиці з методом ітерацій та методом Ньютона розв’язування систем нелінійних рівнянь.

За допомогою цих знань ми реалізували програму, що знайшла розв’язок наступної системи:

$$\begin{cases} \cos(x + 0{,}5) + y = 0{,}8 \\ \sin y - 2x = 1{,}6 \end{cases}$$

Перед цим, ми провели деякі обчислення вручну, які потім використали у програмній реалізації.

Ми отримали вектор-стовпець дійсних чисел, що відображає розв' язок цієї системи:

$$X \approx \begin{pmatrix} -0{,}867 \\ -0{,}134 \end{pmatrix}$$

і довели, що цей результат є вірним.