**Лабораторна робота №2**

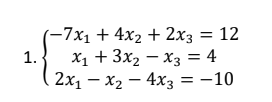
Тема: Принцип стискальних відображень

Мета: Ознайомитися з принципом стискальних відображень, методом його доведення та

деякими прикладами застосування

Завдання 1. Використовуючи метод простої ітерації, розв’язати систему лінійних рівнянь

Варіант:



Крок 1. Записати систему у вигляді xi = ∑ cijxj + di

Перше рівняння:

-7x1 + 4x­2 + 2x­3 = 12

x1 = -12/7 + (4/7)x2 + (2/7)x3­

Друге рівняння:

x1 + 3x2 – x3 = 4

x2 = 4/3 – (1/3)x1 + (1/3)x3

Третє рівняння:

2x1 – x2 - 4x3 = -10

x3 = 10/4 + (2/4)x1 – (1/4)x2 = 2.5 + 0.5x1 – 0.25x2

Отже, система в ітераційному вигляді:

Матриця С та вектор d (переведемо одразу дроби в десяткові значення для зручності):

Крок 2. Перевірити для матриці C отриманої системи умову збіжності ітераційного процесу:

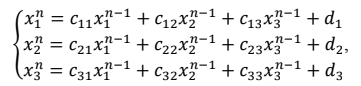
Рядок 1: |4/7| + |2/7| = 4/7 + 2/7 = 6/7 = 0.857 < 1 (Умова виконується)

Рядок 2: |-1/3| + |1/3| = 1/3 + 1/3 = 2/3 = 0.667 < 1 (Умова виконується)

Рядок 3: |1/2| + |-1/4| = 1/2 + 1/4 = 3/4 = 0.75 < 1 (Умова виконується)

Оскільки для кожного рядка сума модулів елементів менша за 1, умова збіжності виконується, і ми можемо застосовувати метод простих ітерацій.

Крок 3. Вибрати довільне початкове значення і знайти наступну ітерацію, виконуючи обчислення вручну (навести повністю всі обрахунки) за рекурентною формулою:



де n позначає номер ітерації.

Нехай початкове наближення

Перша інтеграція (n = 1):

Підставляємо значення початкового наближення:

Отже, х(1) =

Друга ітерація (n = 2):

Підставляємо значення першої інтеграції:

Отже, х(2) =

Крок 4. Написати програму, яка обчислює розв’язок заданої системи із заданою точністю ε = 0.001, тобто реалізувати алгоритм, наведений у кроках 1-3. При цьому програма має знаходити ітерації до того моменту, поки не буде досягнута задана точність. Вхідні дані для програми – коефіцієнти системи і точність обчислень.

using System;

namespace Lab\_2

{

class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = System.Text.Encoding.Unicode;

double[,] C = {

{ 0, 4.0 / 7.0, 2.0 / 7.0 },

{ -1.0 / 3.0, 0, 1.0 / 3.0 },

{ 0.5, -0.25, 0 }

};

double[] d = { -12.0 / 7.0, 4.0 / 3.0, 5.0 / 2.0 };

int n = 3;

double epsilon = 0.001;

// Початкове наближення x^(0)

double[] x\_current = new double[n]; // Всі елементи ініціалізуються нулями за замовчуванням

double[] x\_prev = new double[n];

int iteration = 0;

double maxDiff = double.MaxValue;

Console.WriteLine("Метод простих ітерацій для системи:");

Console.WriteLine("-7x1 + 4x2 + 2x3 = 12");

Console.WriteLine("x1 + 3x2 - x3 = 4");

Console.WriteLine("2x1 - x2 - 4x3 = -10\n");

Console.WriteLine($"Початкове наближення x^{(0)} = [{x\_current[0]:F4}, {x\_current[1]:F4}, {x\_current[2]:F4}]");

while (maxDiff >= epsilon)

{

iteration++;

Array.Copy(x\_current, x\_prev, n); // Зберігаємо попередні значення

for (int i = 0; i < n; i++)

{

double sum = 0;

for (int j = 0; j < n; j++)

{

sum += C[i, j] \* x\_prev[j];

}

x\_current[i] = sum + d[i];

}

// Обчислення максимальної різниці для умови зупинки

maxDiff = 0;

for (int i = 0; i < n; i++)

{

double diff = Math.Abs(x\_current[i] - x\_prev[i]);

if (diff > maxDiff)

{

maxDiff = diff;

}

}

Console.WriteLine($"\nІтерація {iteration}:");

Console.WriteLine($"x^{(iteration)} = [{x\_current[0]:F4}, {x\_current[1]:F4}, {x\_current[2]:F4}]");

Console.WriteLine($"Максимальна різниця між ітераціями: {maxDiff:F6}");

}

Console.WriteLine("\n------------------------------------------------");

Console.WriteLine($"Розв'язок системи з точністю {epsilon} досягнуто за {iteration} ітерацій:");

Console.WriteLine($"x1 = {x\_current[0]:F4}");

Console.WriteLine($"x2 = {x\_current[1]:F4}");

Console.WriteLine($"x3 = {x\_current[2]:F4}");

}

}

}