Звіт

з лабораторної роботи №3

з курсу алгоритми і методи обчислень.

(ІТЕРАЦІЙНІ МЕТОДИ ЯКОБІ ТА ГАУСА – ЗЕЙДЕЛЯ)

Студента 1-го курсу,

Парфенюка Тимофія Павловича

2019

Завдання:

З’ясувати факт збіжності чи розбіжності ітераційних процесів Якобі та Зейделя. У випадку збіжності знайти розв’язок СЛАР з точністю 0.0001 та перевірити його, підставляючи в СЛАР

отримані розв’язки і обраховуючи нев’язки. Визначити порядок збіжності ітераційного процесу.

Код:

using System;

namespace Algoritmes3

{

class Program

{

static public double InputDouble(string s)

{

double input = 0;

string cont = "";

do

{

cont = "";

try

{

Console.WriteLine(s);

input = double.Parse(Console.ReadLine());

}

catch (Exception e)

{

Console.Beep();

Console.WriteLine(e.Message + "Again?");

cont = Console.ReadLine();

}

} while (cont == "yes");

return input;

}

static public double[,] CreateArr()

{

double[,] Arr = new double[3, 4];

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 4; j++)

{

Arr[i, j] = InputDouble("Input [" + i + "," + j + "]-th element");

}

}

return Arr;

}

static public void PrintArray(double[,] Mass, string Message)

{

Console.WriteLine(Message);

for (int i = 0; i < Mass.GetLength(0); i++)

{

for (int j = 0; j < Mass.GetLength(1); j++)

{

Console.Write("{0:f4} ", Mass[i, j]);

if (j == 2)

Console.Write("|");

}

Console.WriteLine();

}

}

static public bool ArrayTest(double[,] arr)

{

if (Math.Abs(arr[0, 1]) + Math.Abs(arr[0, 2]) < Math.Abs(arr[0, 0]))

{

if (Math.Abs(arr[1, 0]) + Math.Abs(arr[1, 2]) < Math.Abs(arr[1, 1]))

{

if (Math.Abs(arr[2, 0]) + Math.Abs(arr[2, 1]) < Math.Abs(arr[2, 2]))

{

Console.WriteLine("Умова збiжностi виконується. Вітаю!");

return true;

}

}

}

Console.WriteLine("Умова збыжності не виконується");

return false;

}

public static void JacobiMethod(double[,] arr, double eps)

{

int lich = 0;

int max = 100;

double x0\_1 = 0;

double x0\_2 = 0;

double x0\_3 = 0;

double x1 = 100;

double x2 = 100;

double x3 = 100;

double b1 = arr[0, 3];

double b2 = arr[1, 3];

double b3 = arr[2, 3];

do

{

lich++;

x0\_1 = x1;

x0\_2 = x2;

x0\_3 = x3;

x1 = (b1 - (arr[0, 1] \* x0\_2) - (arr[0, 2] \* x0\_3)) / arr[0, 0];

x2 = (b2 - (arr[1, 0] \* x0\_1) - (arr[1, 2] \* x0\_3)) / arr[1, 1];

x3 = (b3 - (arr[2, 0] \* x0\_1) - (arr[2, 1] \* x0\_2)) / arr[2, 2];

Console.WriteLine("|k =\t\t|{0}", lich);

Console.WriteLine("|X1k =\t\t|{0}", x1);

Console.WriteLine("|X2k =\t\t|{0}", x2);

Console.WriteLine("|X3k =\t\t|{0}", x3);

Console.WriteLine("||X1k - X1k-1|=\t|{0}", Math.Abs(x1 - x0\_1));

Console.WriteLine("||X2k - X2k-1|=\t|{0}", Math.Abs(x2 - x0\_2));

Console.WriteLine("||X3k - X3k-1|=\t|{0}", Math.Abs(x3 - x0\_3));

Console.WriteLine("| r1 = \t\t|{0}", arr[0, 0] \* x1 + arr[0, 1] \* x2 + arr[0, 2] \* x3 - arr[0, 3]);

Console.WriteLine("| r2 = \t\t|{0}", arr[1, 0] \* x1 + arr[1, 1] \* x2 + arr[1, 2] \* x3 - arr[1, 3]);

Console.WriteLine("| r3 = \t\t|{0}", arr[2, 0] \* x1 + arr[2, 1] \* x2 + arr[2, 2] \* x3 - arr[2, 3]);

Console.WriteLine();

if (lich >= max)

Console.WriteLine("Умова збiжностi простих iтерацiй не виконується, отже метод розбiгається");

} while ((lich <= max) && (Math.Abs(x1 - x0\_1) + Math.Abs(x2 - x0\_2) + Math.Abs(x3 - x0\_3) > eps));

Console.WriteLine("x1 = {0}, x2 = {1}, x3 = {2}", x1, x2, x3);

}

public static void SeidelMethod(double[,] arr, double eps)

{

int lich = 0;

int max = 100;

double x0\_1 = 0;

double x0\_2 = 0;

double x0\_3 = 0;

double x1 = arr[0, 3];

double x2 = arr[1, 3];

double x3 = arr[2, 3];

double b1 = arr[0, 3];

double b2 = arr[1, 3];

double b3 = arr[2, 3];

do

{

lich++;

x0\_1 = x1;

x0\_2 = x2;

x0\_3 = x3;

x1 = (b1 - (arr[0, 1] \* x0\_2) - (arr[0, 2] \* x0\_3)) / (arr[0, 0]);

x2 = (b2 - (arr[1, 0] \* x1) - (arr[1, 2] \* x0\_3)) / (arr[1, 1]);

x3 = (b3 - (arr[2, 0] \* x1) - (arr[2, 1] \* x2)) / (arr[2, 2]);

Console.WriteLine("|k =\t\t|{0}", lich);

Console.WriteLine("|X1k =\t\t|{0}", x1);

Console.WriteLine("|X2k =\t\t|{0}", x2);

Console.WriteLine("|X3k =\t\t|{0}", x3);

Console.WriteLine("||X1k - X1k-1|=\t|{0}", Math.Abs(x1 - x0\_1));

Console.WriteLine("||X2k - X2k-1|=\t|{0}", Math.Abs(x2 - x0\_2));

Console.WriteLine("||X3k - X3k-1|=\t|{0}", Math.Abs(x3 - x0\_3));

Console.WriteLine("| r1 = \t\t|{0}", arr[0, 0] \* x1 + arr[0, 1] \* x2 + arr[0, 2] \* x3 - arr[0, 3]);

Console.WriteLine("| r2 = \t\t|{0}", arr[1, 0] \* x1 + arr[1, 1] \* x2 + arr[1, 2] \* x3 - arr[1, 3]);

Console.WriteLine("| r3 = \t\t|{0}", arr[2, 0] \* x1 + arr[2, 1] \* x2 + arr[2, 2] \* x3 - arr[2, 3]);

Console.WriteLine();

if (lich >= max)

Console.WriteLine("Умова збiжностi простих iтерацiй не виконується, отже метод розбiгається");

} while ((lich <= max) && (Math.Abs(x1 - x0\_1) + Math.Abs(x2 - x0\_2) + Math.Abs(x3 - x0\_3) > eps));

Console.WriteLine("x1 = {0}, x2 = {1}, x3 = {2}", x1, x2, x3);

}

static void Main(string[] args)

{

double eps = Math.Pow(10, -4);

double[,] Array = CreateArr();

PrintArray(Array, "Матриця: Ax = b --->>");

if (ArrayTest(Array))

{

JacobiMethod(Array, eps);

//SeidelMethod(Array, eps);

}

}

}

}

***Висновок:***

*Умова збіжності методів – діагональна перевага. Для цього треба було перетворити рівняння, щоб матриця коефіцієнтів мала діагонльну перевагу.*

Метод Зейделя має більшу швидкість збіжності, ніж метод Якобі.

***Вхідні та вихідні данні:***

