**Расчетно-графическая работа №6**

**Тема: «Численные методы решения краевой задачи для обыкновенных дифференциальных уравнений»**

Для уравнения с краевыми условиями решить краевую задачу методом стрельбы

,

и методом прогонки

,

.

Решение задачи Коши в методе стрельбы: *Метод Эйлера*

Решение нелинейного уравнения в методе стрельбы: *Метод Ньютона*

Аппроксимация краевых условий в методе прогонки: *по двум узлам*

1. **Точное решение**

1) Однородное уравнение:

2) Неоднородное уравнение: Решение будем находить в виде частного решения

A = 2.1 B = -2.1 C = -2

3) Общее решение:

Воспользуемся краевыми условиями для нахождения

4) Точное решение:

**2. Метод стрельбы**

= *f(x,y)*

Введем ещё одну замену (нужна она для того, чтобы найти параметр µ)

Находим значения

Хотим получить, чтобы

Если не получается, тогда

Находим методом Ньютона:

Найдем нужные методом Эйлера и подставим в Условие выхода

**3. Метод прогонки. По двум узлам**

,

,

.

Матрица трех диагональная и диагональное преобладание выполнено

Начнем разбираться с левым краевым условием:

Диагональное преобладание выполнено

Теперь будем разбираться с правым краевым условием:

Из второго уравнения вычтем первое:

Диагональное преобладание выполнено

Введем замену

Находим все

Подставляем в

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Solution\_of\_the\_boundary\_value\_problem\_for\_ODE

{

class Program

{

public static double FY(double x, double z)

{

return z;

}

public static double FZ(double x, double y)

{

return (y + 6.2 + 2.1 \* x \* (1 - x));

}

public static double FU(double x, double v)

{

return v;

}

public static double FV(double x, double u)

{

return u;

}

public static double EulersForY(double n, double x, double z)

{

var h = 1 / n;

return (h \* FY(x, z));

}

public static double EulersForZ(double n, double x, double y)

{

var h = 1 / n;

return (h \* FZ(x, y));

}

public static double FuncZ(double n, double m)

{

var y = 0.0;

var z = m;

var h = 1 / n;

var x = 0.0;

Console.WriteLine("x = " + x + " y = " + y + " z = " + z);

for (int i = 1; i <= n; i++)

{

z = z + EulersForZ(n, x, y);

y = y + EulersForY(n, x, z);

x += h;

Console.WriteLine("x = " + x + " y = " + y + " z = " + z);

}

return z;

}

public static double g(double n, double m)

{

return (FuncZ(n, m) - Math.E + (1.0 / Math.E) - 2.1);

}

public static double EulersForV(double n, double x, double u)

{

var h = 1 / n;

return (h \* FV(x, u));

}

public static double EulersForU(double n, double x, double v)

{

var h = 1 / n;

return (h \* FU(x, v));

}

public static double FuncV(double n, double m)

{

var v = 0.0;

var u = 1.0;

var h = 1 / n;

var x = 0.0;

for (int i = 1; i <= n; i++)

{

v = v + EulersForV(n, x, u);

u = u + EulersForU(n, x, v);

x += h;

}

return v;

}

public static double Shooting(double n, double m)

{

var E = 0.00005;

int i = 1;

FuncZ(n, m);

var m1 = m - (g(n, m) / FuncV(n, m));

FuncZ(n, m1);

while (Math.Abs(m - m1) > E)

{

m = m1;

m1 = m - (g(n, m) / FuncV(n, m));

i++;

FuncZ(n, m1);

}

return m1;

}

public static void Running(int n)

{

double[,] znach = new double[10, 2];

var h = 1.0 / n;

var N11 = 0.0;

var N12 = 0.0;

znach[0, 0] = N11;

znach[0, 1] = N12;

Console.WriteLine("Ля 0 = " + znach[0, 0] + " Мю 0 = " + znach[0, 1]);

var x = 0.0 + h;

for (int i = 1; i < n; i++)

{

znach[i, 0] = -1 / (N11 - 2 - Math.Pow(h, 2));

znach[i, 1] = ((6.2 + 2.1 \* x \* (1 - x)) \* Math.Pow(h, 2) - N12) / (N11 - 2 - Math.Pow(h, 2));

N11 = znach[i, 0];

N12 = znach[i, 1];

x += h;

Console.WriteLine("Ля " + i + " = " + znach[i, 0] + " Мю " + i + " = " + znach[i, 1]);

}

var yn = 1.0 / (1 + Math.Pow(h, 2)) \* ((6.2 \* Math.Pow(h, 2) - (Math.E - 1.0 / Math.E + 2.1) \* h) / (znach[n-1, 0] - 1 - Math.Pow(h, 2)) +(znach[n-1, 1] \* (1 + Math.Pow(h, 2))) / (1 + Math.Pow(h, 2) - znach[n-1,0])) + (6.2 \* Math.Pow(h, 2) - (Math.E - 1.0 / Math.E + 2.1) \* h) / (-1 - Math.Pow(h, 2));

Console.WriteLine("x = " + 1 + " y = " + yn);

for (int i = n; i > 0; i--)

{

var yi = yn \* znach[i-1, 0] + znach[i-1, 1];

Console.WriteLine("x = " + (i - 1) + " y = " + yi);

yn = yi;

}

}

static void Main(string[] args)

{

Shooting(10, 1);

Running(10);

Shooting(20, 1);

Running(20);

}

}

}