МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФИЛИАЛ ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «МЭИ» В Г. СМОЛЕНСКЕ

Кафедра вычислительной техники

Отчет по лабораторной работе №5

Тема: «Поиск минимума многомерной функции методом Хука-Дживса».

Группа: ИВТ1-18

Студент: Данилов И. В.

Преподаватель: Гаврилов А.И.

Вариант: № 4

Смоленск 2020 г.

**Цель работы**

Цель работы – изучение метода Хука-Дживса, применяемого для поиска экстремума многомерной функции.

**Задание**

Определить с помощью метода Хука-Дживса точку минимума функций.

Функция:

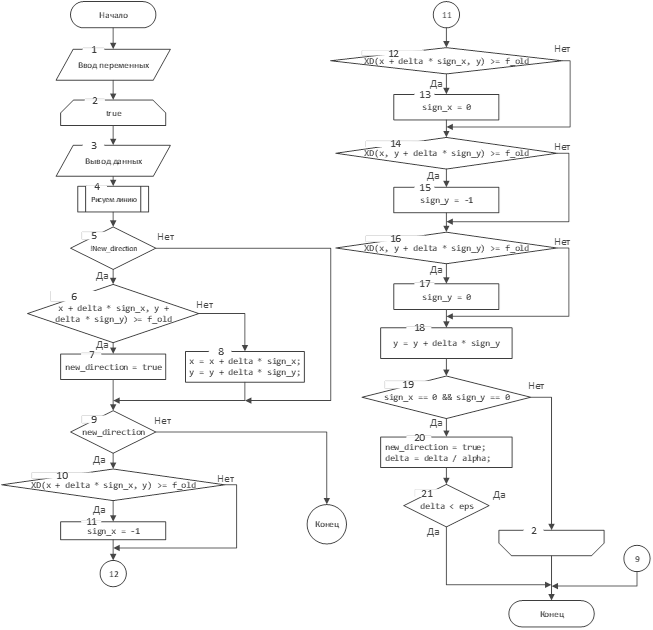
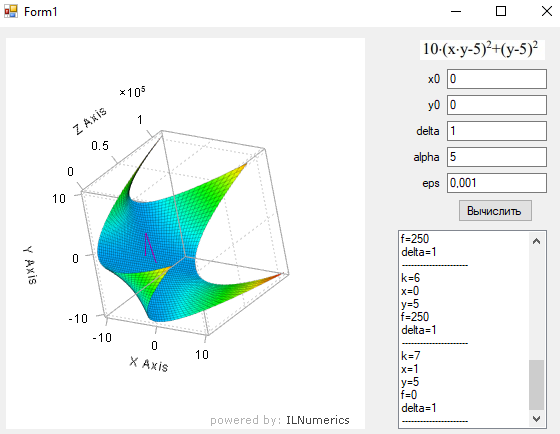
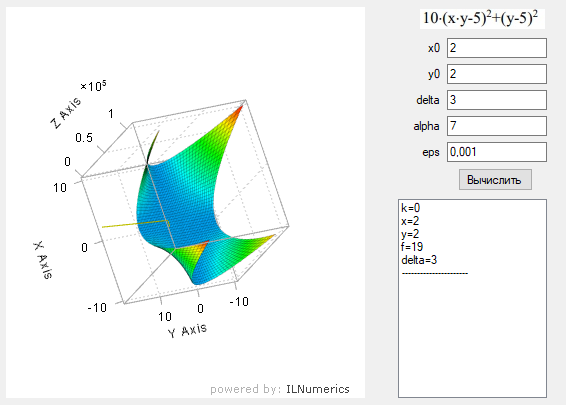
**Блок-схема алгоритма**

Рисунок 1 – алгоритм программы

**Тесты**





**Код программы**

using ILNumerics.Drawing;

using ILNumerics.Drawing.Plotting;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private double XD(double x, double y) =>

10 \* Math.Pow(x \* y - 5, 2) + Math.Pow(y - 5, 2);

private void DrawLine(double xx, double yy, double f)

{

var scene = new Scene()

{

new PlotCube(twoDMode: false)

{

new Surface(

(x,y)=> (float)(10\*Math.Pow(x\*y - 5, 2) + Math.Pow(y-5,2))),

new LinePlot(xx,yy,f)

}

};

panel1.Scene = scene;

}

private void button1\_Click(object sender, EventArgs e)

{

Info.Items.Clear();

var x0 = Convert.ToDouble(x0Box.Text);

var y0 = Convert.ToDouble(y0Box.Text);

var delta = Math.Abs(Convert.ToDouble(deltaBox.Text));

var alpha = Math.Abs(Convert.ToDouble(alphaBox.Text));

var eps = Math.Abs(Convert.ToDouble(epsBox.Text));

var x = x0;

var y = x0;

var x\_min = x0;

var y\_min = y0;

var new\_direction = true;

var sign\_x = 1;

var sign\_y = 1;

var k = 0;

double f\_old;

double x\_old, y\_old;

while (true)

{

f\_old = XD(x, y);

AddInfo(k, x, y, f\_old, delta);

x\_old = x;

y\_old = y;

DrawLine(x\_old, y\_old, f\_old);

if (!new\_direction)

{

if (XD(x + delta \* sign\_x, y + delta \* sign\_y) >= f\_old)

{

new\_direction = true;

}

else

{

x = x + delta \* sign\_x;

y = y + delta \* sign\_y;

}

}

if (new\_direction)

{

sign\_x = 1;

if (XD(x + delta \* sign\_x, y) >= f\_old)

{

sign\_x = -1;

}

if (XD(x + delta \* sign\_x, y) >= f\_old)

{

sign\_x = 0;

}

sign\_y = 1;

if (XD(x, y + delta \* sign\_y) >= f\_old)

{

sign\_y = -1;

}

if (XD(x, y + delta \* sign\_y) >= f\_old)

{

sign\_y = 0;

}

y = y + delta \* sign\_y;

new\_direction = false;

if (sign\_x == 0 && sign\_y == 0)

{

new\_direction = true;

delta = delta / alpha;

if (delta < eps)

{

break;

}

}

}

DrawLine(x, y, f\_old);

k++;

}

panel1.Refresh();

}

private void AddInfo(int k, double x, double y, double f\_old, double delta)

{

Info.Items.Add("k=" + k);

Info.Items.Add("x=" + x);

Info.Items.Add("y=" + y);

Info.Items.Add("f=" + f\_old);

Info.Items.Add("delta=" + delta);

Info.Items.Add("----------------------");

}

private void eps\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == '.')

e.KeyChar = ',';

}

}

}