МИНИСТЕРСТВО ВЫСШЕГО ОБРАЗОВАНИЯ И НАУКИ РФ

филиал федерального государственного бюджетного образовательного учреждения высшего образования «Национальный исследовательский университет «МЭИ» в г. Смоленске

Кафедра вычислительной техники

Отчет по лабораторной работе №6

по курсу: «Введение в оптимизацию»

Тема: «ПОИСК МИНИМУМА МНОГОМЕРНОЙ ФУНКЦИИ МЕТОДОМ НЕЛДЕРА-МИДА»

Группа: ИВТ1-18

Студент: \*\*\*

Преподаватель: Гаврилов А. И.

Вариант: № 3

Смоленск, 2020 г.

1. **Цель работы**

Цель работы – изучение метода Нелдера-Мида, применяемого для поиска экстремума многомерной функции..

1. **Задание**

Определить с помощью метода Нелдера-Мида точку минимума функций.

Функция:



1. **Схема алгоритма.**



Рисунок 1 – Алгоритм программы

1. **Спецификация программы, раскрывающая смысл входных и выходных данных, основных переменных и функций.**
2. **Результаты тестирования программы на наборе целевых функций с указанием числа итераций и количества вычислений целевой функции. Таблица, иллюстрирующая итерации вычислительного процесса и изменение ключевых переменных.**

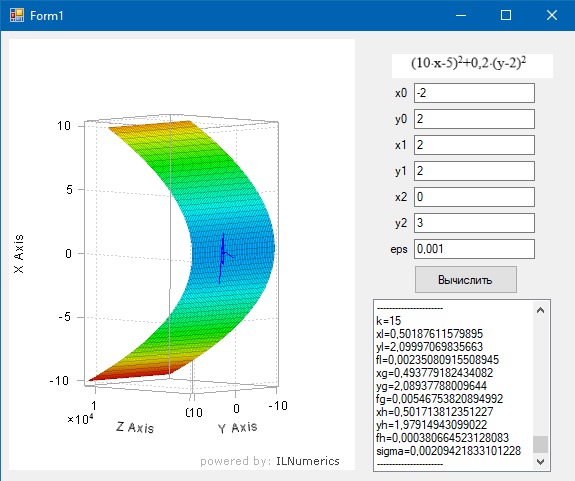


Рисунок 2 – Результат выполнения программы

1. **Графическая интерпретация процесса оптимального поиска на поле графика целевой функции, реализованного на языке C#.**

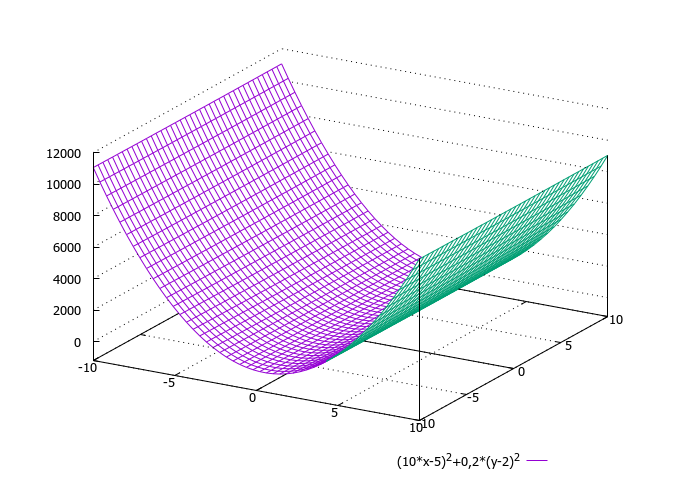
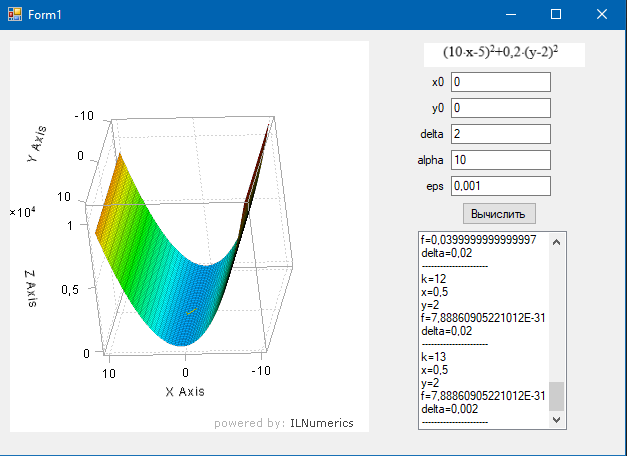


Рисунок 3 – График функции

1. **Выводы по работе.**

В результате выполнения лабораторной работы были изучен метод Нелдера-Мид, применяемого для поиска экстремума многомерной функции.

Значения приблизительно равны, как и в методе Хука-Дживса



1. **Текст программы с комментариями.**

using ILNumerics;

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 \_5.\_1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

Math.Pow(x - 2, 2)

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

+ (0.5 \* Math.Pow(y + 1, 2));

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

{

var scene = new Scene()

{

new PlotCube(twoDMode: false)

{

new Surface(

(x,y)=> (float)(Math.Pow(x - 2, 2)

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

+ (0.5 \* Math.Pow(y + 1, 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;

//var app = new Microsoft.Office.Interop.Word.Application();

//app.Visible = false;

//var doc = app.Documents.Add();

//doc.Tables.Add(doc.Range(0, 0), 100, 11);

while (true)

{

f\_old = XD(x, y);

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

//WordTable(doc, k, x, y, f\_old, delta);

//От

x\_old = x;

y\_old = y;

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

//По старому

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;

}

}

//По новому

if (new\_direction)

{

//x

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;

}

//y

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;

//Меняем

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();

//app.Visible = true;

}

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 = ',';

}

private void WordTable(Microsoft.Office.Interop.Word.Document doc,

int k, double x, double y, double f\_old, double delta)

{

try

{

doc.Tables[1].Cell(k, 1).Range.InsertAfter(k.ToString());

doc.Tables[1].Cell(k, 2).Range.InsertAfter(Math.Round(x, 3).ToString());

doc.Tables[1].Cell(k, 3).Range.InsertAfter(Math.Round(y, 3).ToString());

doc.Tables[1].Cell(k, 4).Range.InsertAfter(Math.Round(f\_old, 3).ToString());

doc.Tables[1].Cell(k, 5).Range.InsertAfter(Math.Round(delta, 3).ToString());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

}