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 №4

по курсу: «Введение в оптимизацию»

Тема: «ОДНОМЕРНЫЙ ПОИСК МИНИМУМА УНИМОДАЛЬНЫХ ФУНКЦИЙ МЕТОДОМ ПАУЭЛЛА»

Группа: ИВТ1-18

Студент: Гоголь Н. М.

Преподаватель: Гаврилов А. И.

Вариант: № 3

Смоленск, 2020 г.

1. **Цель работы**

Цель работы – изучение метода квадратичной интерполяции – метода Пауэлла для минимизации функций одной переменной.

1. **Задание**
2. Определить с помощью метода Пауэлла точку минимума функций.

Функции:



1. **Схема алгоритма.**



Рисунок 1 – Алгоритм программы

1. **Спецификация программы, раскрывающая смысл входных и выходных данных, основных переменных и функций.**

X – начальная точка

H – шаг

K – количество итераций

Eps1, eps2 – сравниваемые значения

1. **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.**

Первая функция:

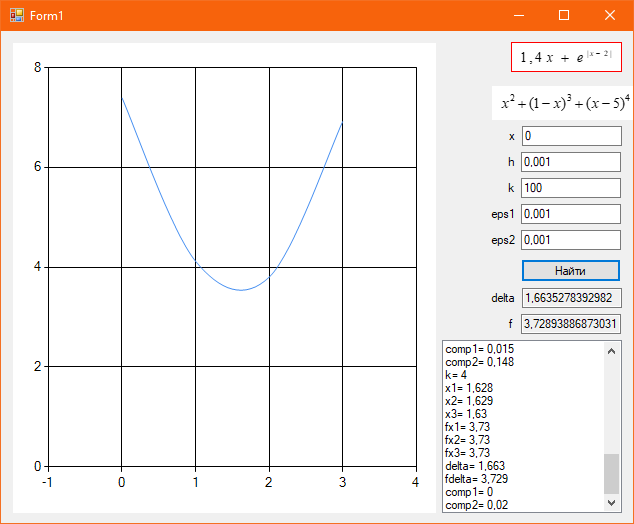


Рисунок 1 – Тестирование первой функции

Вторая функция:

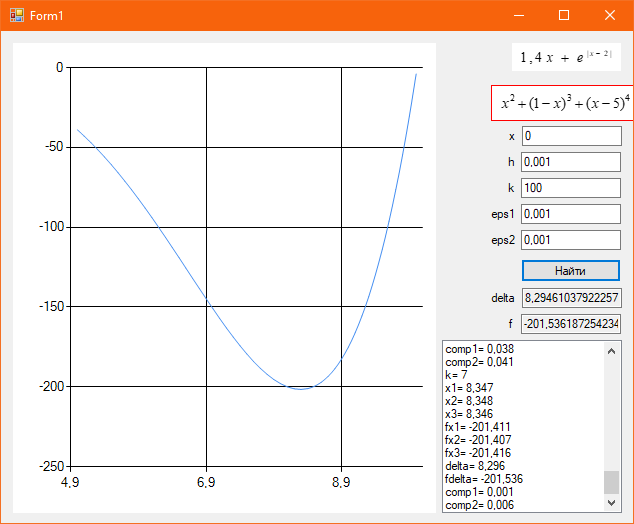


Рисунок 2 – Тестирование второй функции

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

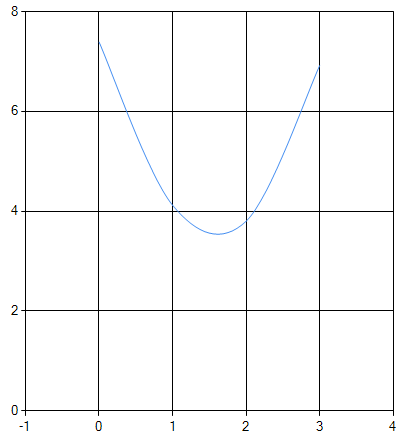


Рисунок 3 – График первой функции

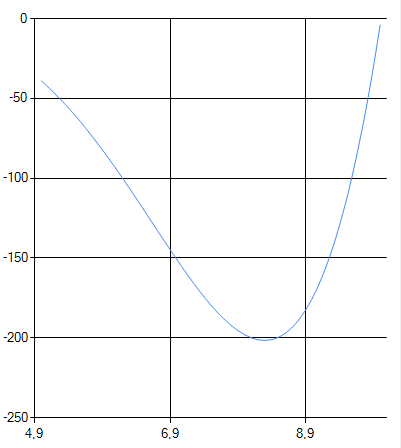
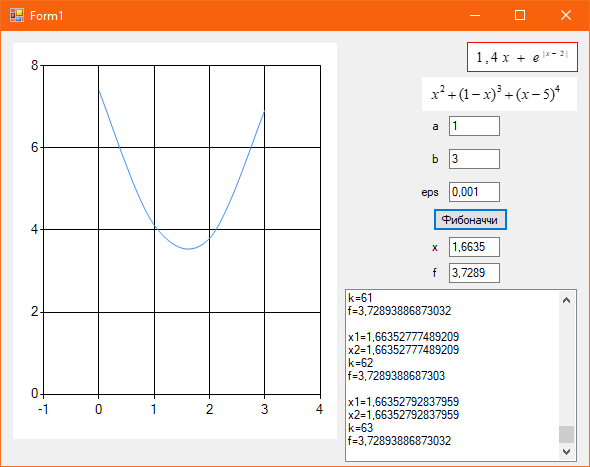


Рисунок 4 - График второй функции

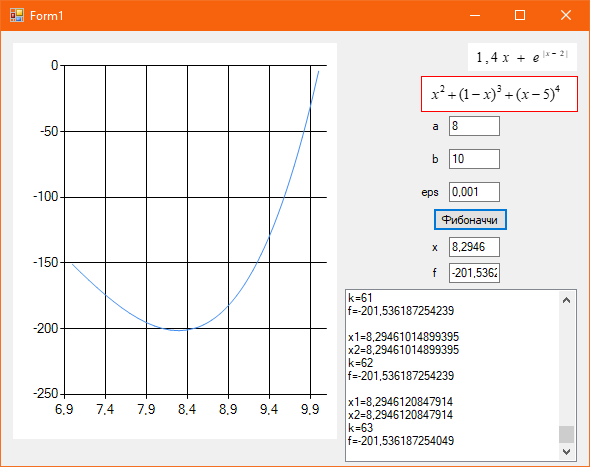
1. **Выводы по работе.**

В результате выполнение лабораторной работы были изучены методызолотого сечения для минимизации функций одной переменной.

Данные сходятся с другими методами поиска, например метод Фибоначчи. Для первой функции



Для второй



1. **Текст программы с комментариями.**

usingSystem;

usingSystem.Drawing;

usingSystem.Windows.Forms;

namespace \_4

{

publicpartialclassForm1 :Form

{

publicForm1()

{

InitializeComponent();

}

privateshortchoiceFunc = 0;

privatevoid pictureBox1\_Click(object sender, EventArgs e)

{

chart1.Series[0].Points.Clear();

choiceFunc = 1;

Refresh();

Draw();

xBox.Text = "0";

hBox.Text = "0,001";

eps1Box.Text = "0,001";

eps2Box.Text = "0,001";

}

privatevoid pictureBox2\_Click(object sender, EventArgs e)

{

chart1.Series[0].Points.Clear();

choiceFunc = 2;

Refresh();

Draw();

xBox.Text = "0";

hBox.Text = "0,001";

eps1Box.Text = "0,001";

eps2Box.Text = "0,001";

}

privatevoidDraw()

{

switch (choiceFunc)

{

case 1:

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

chart1.Series[0].Points.AddXY(i, Funcs(i));

break;

case 2:

for (doublei = 5; i<= 10; i += 0.1)

chart1.Series[0].Points.AddXY(i, Funcs(i));

break;

default:

break;

}

}

privatedoubleFuncs(double x)

{

switch (choiceFunc)

{

case 1:

return 1.4 \* x + Math.Pow(Math.E, Math.Abs(x - 2));

case 2:

returnMath.Pow(x, 2) + Math.Pow(1 - x, 3) + Math.Pow(x - 5, 4);

default:

break;

}

return -1;

}

privatevoidSearch\_Button\_Click(object sender, EventArgs e)

{

if (choiceFunc == 0)

{

MessageBox.Show("Выберитефункцию", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

listBox1.Items.Clear();

try

{

//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);

var x0 = Convert.ToDouble(xBox.Text); // x0

var h = Math.Abs(Convert.ToDouble(hBox.Text)); // h

var eps1 = Math.Abs(Convert.ToDouble(eps1Box.Text)); // eps1

var eps2 = Math.Abs(Convert.ToDouble(eps2Box.Text)); // eps2

var steps = Convert.ToInt32(kBox.Text);

var x1 = x0;

var fx1 = Funcs(x1);

double fx2, fx3, // f

delta, fdelta,

xmin, fxmin,

comp1, comp2,

x2, x3,

total = 0;

int k = 1; //счетчик

while (true)

{

//Вторая точка

x2 = x1 + h;

fx2 = Funcs(x2);

//Третья точка

if (fx1 > fx2)

x3 = x1 + 2 \* h;

else

x3 = x1 - h;

fx3 = Funcs(x3);

//Минимум

xmin = x1;

fxmin = fx1;

if (fx2 <fxmin)

{

xmin = x2;

fxmin = fx2;

}

if (fx3 <fxmin)

{

xmin = x3;

fxmin = fx3;

}

//Дельта

var d1 = (x1 + x2) / 2;

var d2 = ((fx1 - fx2) \* (x2 - x3) \* (x3 - x1) / ((x2 - x3) \* fx1 + (x3 - x1) \* fx2 + (x1 - x2) \* fx3));

var d3 = d2 / 2;

delta = d1 + d3;

//delta = (x1 + x2) / 2 + (fx1 - fx2) \* (x2 - x3) \* (x3 - x1) / ((x2 - x3) \* fx1 + (x3 - x1) \* fx2 + (x1 - x3) \* fx3);

fdelta = Funcs(delta);

//Проверка

comp1 = Math.Abs((fxmin - fdelta) / fdelta);

comp2 = Math.Abs((xmin - delta) / delta);

if (comp1 < eps1 && comp2 < eps2)

{

total = delta;

break;

}

listBox1.Items.Add($"k= {k}");

listBox1.Items.Add($"x1= {Math.Round(x1, 3)}");

listBox1.Items.Add($"x2= {Math.Round(x2, 3)}");

listBox1.Items.Add($"x3= {Math.Round(x3, 3)}");

listBox1.Items.Add($"fx1= {Math.Round(fx1, 3)}");

listBox1.Items.Add($"fx2= {Math.Round(fx2, 3)}");

listBox1.Items.Add($"fx3= {Math.Round(fx3, 3)}");

listBox1.Items.Add($"delta= {Math.Round(delta, 3)}");

listBox1.Items.Add($"fdelta= {Math.Round(fdelta, 3)}");

listBox1.Items.Add($"comp1= {Math.Round(comp1, 3)}");

listBox1.Items.Add($"comp2= {Math.Round(comp2, 3)}");

listBox1.TopIndex = listBox1.Items.Count - 1;

//WordTable(doc, k, x1, x2, x3, fx1, fx2, fx3, delta, fdelta, comp1, comp2);

//Заменяемточки

if (fdelta<fxmin || x1 == xmin)

{

x1 = delta;

fx1 = fdelta;

}

else

{

x1 = xmin;

fx1 = fxmin;

}

//Выход

if (steps > 1)

steps--;

else

{

listBox1.Items.Add("Превышен k");

total = delta;

break;

}

k++;

}

deltaBox.Text = delta.ToString();

fBox.Text = fdelta.ToString();

listBox1.TopIndex = listBox1.Items.Count - 1;

//app.Visible = true;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

privatevoid Dot\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == '.')

{

e.KeyChar = ',';

}

}

privatevoid Form1\_Paint(object sender, PaintEventArgs e)

{

if (choiceFunc == 0)

return;

Graphics g = e.Graphics;

Pen pen = new Pen(Color.Red);

switch (choiceFunc)

{

case 1:

g.DrawRectangle(pen, pictureBox1.Location.X - 1, pictureBox1.Location.Y - 1, pictureBox1.Width + 1, pictureBox1.Height + 1);

break;

case 2:

g.DrawRectangle(pen, pictureBox2.Location.X - 1, pictureBox2.Location.Y - 1, pictureBox2.Width + 1, pictureBox2.Height + 1);

break;

default:

break;

}

g.Dispose();

}

}

}