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 №1

по курсу: «Введение в оптимизацию»

Тема: «ОДНОМЕРНЫЙ ПОИСК МИНИМУМА УНИМОДАЛЬНЫХ ФУНКЦИЙ МЕТОДАМИ РАВНОМЕРНОГО ПОИСКА И ДИХОТОМИИ»

Группа: ИВТ1-18

Студент: ГогольН. M.

Преподаватель: Гаврилов А. И.

Вариант: № 3

Смоленск, 2020 г.

1. **Цель работы**

Изучение методов одномерной минимизации функций одной переменной.

1. **Задание**
2. Используя метод Свенна, локализовать интервал функции, содержащий минимум.
3. Методом равномерного поиска определить минимум заданных функций.
4. Определить с помощью метода дихотомии точки минимума функцийи количество итераций, необходимых для достижения заданной точности е=0,001.

Функции:



1. **Схема алгоритма.**



Рисунок 1 – Алгоритм Свенна



Рисунок 2 – Алгоритм равномерного поиска



Рисунок 3 – Алгоритм дихотомии

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

А – левая граница

B – правая граница

X – точка на текущей итерации

K– номер итерации

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.**

Тестирование Свенна

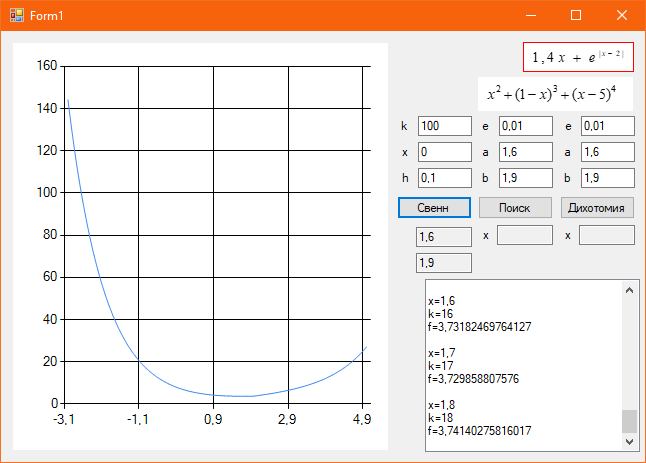


Рисунок 1 – Первый график

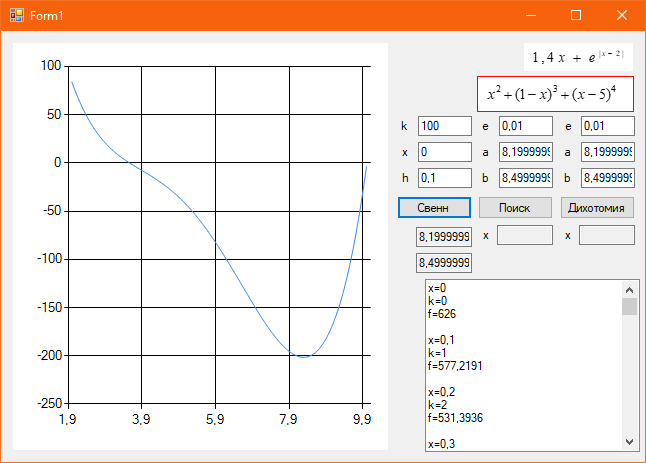


Рисунок 2 – Второй график

Тестирование поиска

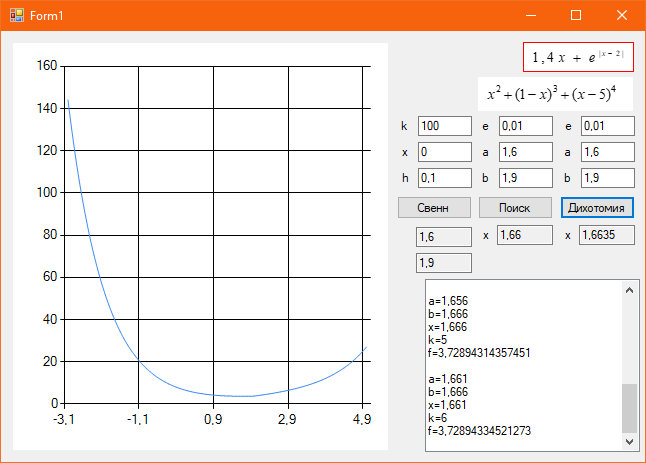


Рисунок 3 – Первый график

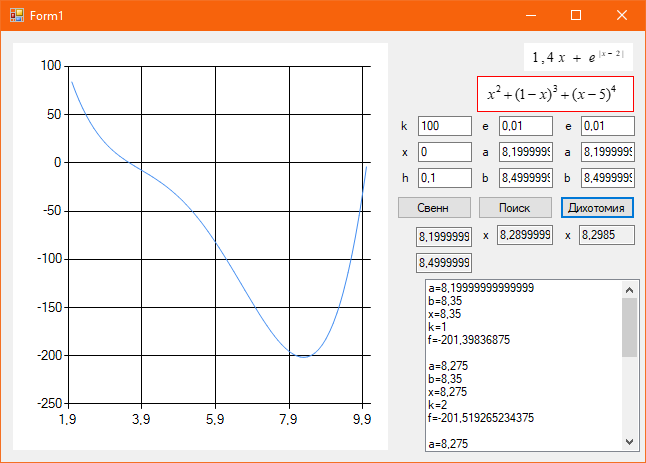


Рисунок 4 – Второй график

Тестирование дихотомии

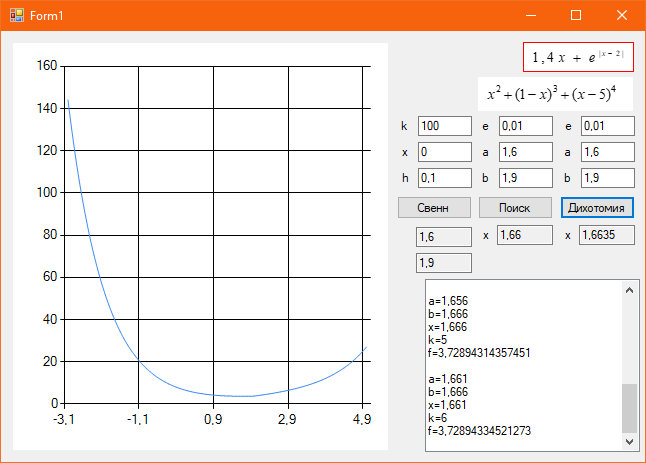


Рисунок 5 -Первый график

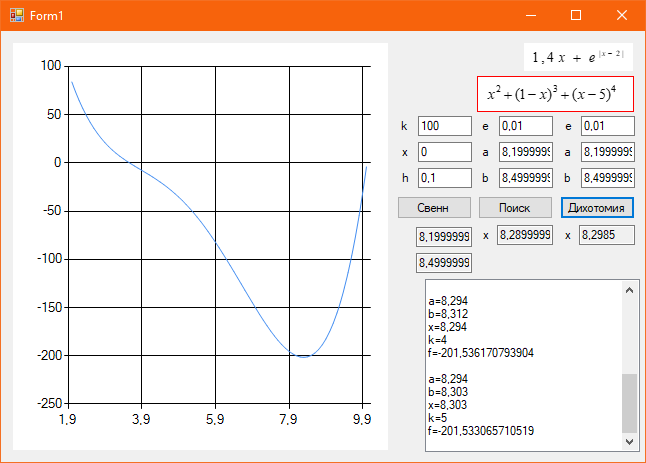


Рисунок 6 – Второй график

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

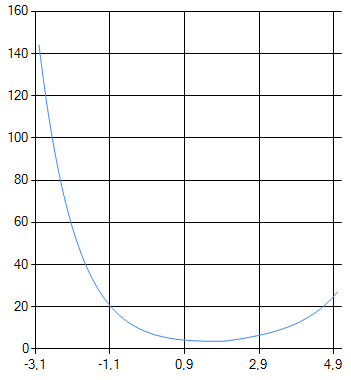


Рисунок 7 – График первой функции

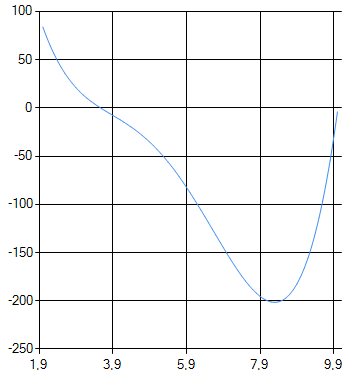


Рисунок 8 - График второй функции

1. **Выводы по работе.**

В результате выполнение лабораторной работы были изучены методы одномерной минимизации функций одной переменной.

1. **Текст программы с комментариями.**

using System;

using System.Drawing;

using System.Windows.Forms;

namespace \_1lab

{

publicpartialclassForm1 : Form

{

publicForm1()

{

InitializeComponent();

}

///<summary>

/// Выбрана ли картинка . 0 - ничего не выбрано / 1 - первая функция / 2 - вторая функция

///</summary>

privateshort choice = 0;

privateint choiceFunc = 0;

privatevoid Form1\_Paint(object sender, PaintEventArgs e)

{

Graphics g = e.Graphics;

Pen pen = new Pen(Color.Red);

switch (choice)

{

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();

}

privatevoid pictureBox1\_Click(object sender, EventArgs e)

{

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

choice = 1;

choiceFunc = 1;

Refresh();

Draw();

}

privatevoid pictureBox2\_Click(object sender, EventArgs e)

{

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

choice = 2;

choiceFunc = 2;

Refresh();

Draw();

}

///<summary>

///МетодСвенна

///</summary>

///<param name="sender"></param>

///<param name="e"></param>

privatevoid SvenButton\_Click(object sender, EventArgs e)

{

if (CheckChoice()) return;

InfoList.Items.Clear();

int k = Convert.ToInt32(kBox.Text);

double x = Convert.ToDouble(xBox.Text);

double h = Convert.ToDouble(hBox.Text);

double xNext = x, a, b;

int i = 0;

bool ch = false;

do

{

if (ch)

{

x += h;

}

InfoList.Items.Add("x=" + x.ToString());

InfoList.Items.Add("k=" + i);

InfoList.Items.Add("f=" + Funcs(x));

InfoList.Items.Add(Environment.NewLine);

if (Funcs(x - h) <= Funcs(x) && Funcs(x + h) >= Funcs(x)) //a

{

a = x - h;

b = x + h;

break;

}

elseif (Funcs(x - h) >= Funcs(x) && Funcs(x + h) <= Funcs(x))//б

{

xNext += h;

}

elseif (Funcs(x - h) <= Funcs(x) && Funcs(x) >= Funcs(x + h))//в

{

h = -1 \* h;

xNext += h;

}

elseif (Funcs(x - h) <= Funcs(x) && Funcs(x + h) <= Funcs(x)) //г

{

MessageBox.Show("11");

return;

}

i++;

ch = true;

} while (Funcs(xNext) <= Funcs(x) && i < k);

if (h > 0)

{

a = x - 2\*h;

b = x + h;

aBox.Text = a.ToString();

bBox.Text = b.ToString();

a2box.Text = a.ToString();

b2box.Text = b.ToString();

a3Box.Text = a.ToString();

b3Box.Text = b.ToString();

}

elseif (h < 0)

{

a = x + h;

b = x - h;

aBox.Text = a.ToString();

bBox.Text = b.ToString();

a2box.Text = a.ToString();

b2box.Text = b.ToString();

a3Box.Text = a.ToString();

b3Box.Text = b.ToString();

}

}

///<summary>

///Функции

///</summary>

///<param name="x">Значение x</param>

///<returns></returns>

privatedouble Funcs(double x)

{

switch (choiceFunc)

{

case 1:

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

case 2:

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

default:

break;

}

return -1;

}

///<summary>

///Рисуетграфики

///</summary>

privatevoid Draw()

{

switch (choiceFunc)

{

case 1:

for (double i = -3; i <= 5; i += 0.1)

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

break;

case 2:

for (double i = 2; i <= 10; i += 0.1)

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

break;

default:

break;

}

}

///<summary>

///Алгоритмравномерногопоиска

///</summary>

///<param name="sender"></param>

///<param name="e"></param>

privatevoid SearchButton\_Click(object sender, EventArgs e)

{

if (CheckChoice()) return;

InfoList.Items.Clear();

double a = Convert.ToDouble(a2box.Text);

double b = Convert.ToDouble(b2box.Text);

double h = Convert.ToDouble(eBox.Text);

double x = a;

int k = 1;

while (k!=100)

{

if (Funcs(x) > Funcs(x + h))

{

x += h;

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

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

InfoList.Items.Add("f=" + Funcs(x));

InfoList.Items.Add(Environment.NewLine);

}

elseif (x > b)

{

InfoList.Items.Add("Точка x на границе интервала " + x);

break; ;

}

else

break;

}

x2Box.Text = x.ToString();

}

///<summary>

/// Заменяет точку на запятую

///</summary>

///<param name="sender"></param>

///<param name="e"></param>

privatevoid Box\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == '.')

e.KeyChar = ',';

}

///<summary>

///Методдихотомии

///</summary>

///<param name="sender"></param>

///<param name="e"></param>

privatevoid DihButton\_Click(object sender, EventArgs e)

{

if (CheckChoice()) return;

InfoList.Items.Clear();

double a = Convert.ToDouble(a3Box.Text);

double b = Convert.ToDouble(b3Box.Text);

double ep = Convert.ToDouble(e2Box.Text);

int k = 1;

do

{

double x = Math.Round((a + b) / 2, 3);

if (Funcs(x - ep / 2) > Funcs(x + ep / 2))

{

a = x;

}

elseif (Funcs(x - ep / 2) < Funcs(x + ep / 2))

{

b = x;

}

InfoList.Items.Add("a=" + a);

InfoList.Items.Add("b=" + b);

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

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

InfoList.Items.Add("f=" + Funcs(x));

InfoList.Items.Add(Environment.NewLine);

} while ((b - a) > ep);

x3Box.Text = ((b + a) / 2).ToString();

}

///<summary>

/// Сделан ли выбор

///</summary>

privatebool CheckChoice()

{

if (choice == 0)

{

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

returntrue;

}

elsereturnfalse;

}

}

}