МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

“ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГИЙ УПРАВЛІННЯ

Лабораторна робота № 2

з курсу «Дослідження операцій»

«Вивчення методів безумовної оптимізації нульового порядку»

Виконав:

студент групи КН 36-а

Кулик Володимир

Перевірив:

Гужва В. О.

ХАРКІВ 2018

Мета роботи:

Вивчити методи безумовної оптимізації нульового порядку, скласти блок-схему алгоритму вибраного методу, скласти програму зі складеного блок-схеми. Необхідно мінімізувати функцію методом Хука-Дживса з використанням одномірної мінімізації.

Завдання:

|  |  |  |  |
| --- | --- | --- | --- |
| Номер варіанта | Вид функції | Характер екстремуму | Початкова умова |
| 8 |  | *min* | (-1,1) |

Алгоритми методу:

Початковий етап.

Вибрати у якості напрямку:.

У нашому випадку двовимірна площина: .

Обрати число для зупинки алгоритму.

Обрати початкову точку . Покласти = та йти до кроку 1.

Основний етап.

**Крок 1**. Обчислити – оптимізаційне рішення задачі мінімізації .

.

Якщо j<n, то замінити j на j+1 та повторити крок 1.

Якщо j=n, то покласти .

Якщо – останов.

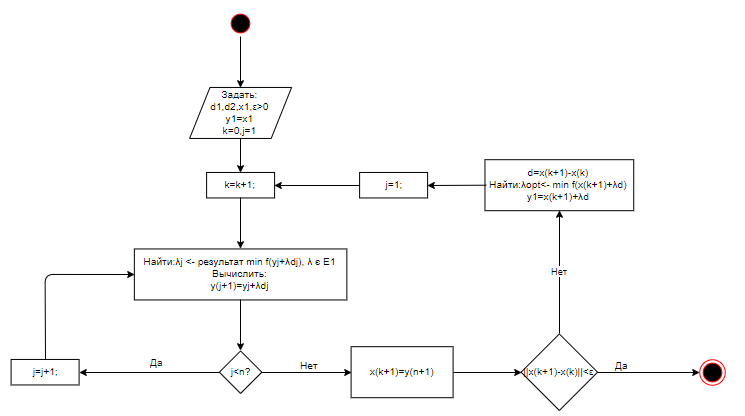
Інакше йти до кроку 2.

**Крок 2**. Покласти та знайти – рішення задачі .

Покласти , j=1.

Замінити k=k+1 та йти до кроку 1.

Блок-схема алгоритму:



Код програми:

Для того, щоб програма працювала необхідно створити проект з формою, як показано на рисунку 1.

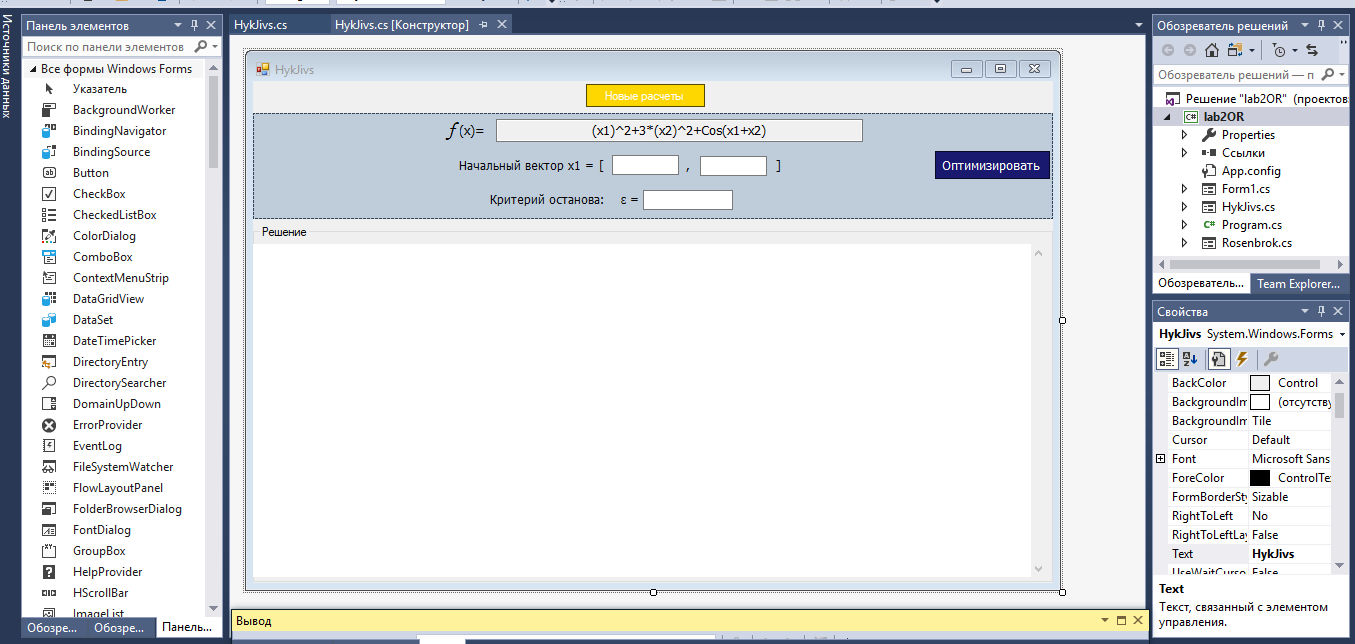


Рисунок 1 – Головна форма проекту

Для того, щоб форма виглядала як на рисунку 1 необхідно з панелі елементів перетягнути необхідні елементи на форму.

Лістинг програми:

this.panel1.Enabled = false;

double x1 = double.Parse(x1Box.Text.Replace(".", ","));

double x2 = double.Parse(x2Box.Text.Replace(".", ","));

double epsel = double.Parse(epsBox.Text.Replace(".", ","));

double f(double[] variable)

{

double res = Math.Pow(variable[0], 3) - variable[0] + Math.Pow(variable[1], 3) - 3 \* Math.Pow(variable[1], 2) - 1;

return res;

}

double[] addVecToVec(double[] var1, double[] var2)

{

double[] res = new double[var1.GetLength(0)];

for (int i = 0; i < var1.GetLength(0); i++)

{

res[i] += var1[i] + var2[i];

}

return res;

}

double[] subtractVecToVec(double[] var1, double[] var2)

{

double[] res = new double[var1.GetLength(0)];

for (int i = 0; i < var1.GetLength(0); i++)

{

res[i] += var1[i] - var2[i];

}

return res;

}

double[] multVector(double[] var1, double m)

{

double[] res = new double[var1.GetLength(0)];

for (int i = 0; i < var1.GetLength(0); i++)

{

res[i] += var1[i] \* m;

}

return res;

}

double func\_minimization(double[] var, double[] d\_mas)

{

double opt\_lyambda, l1, l2, f1, f2;

double epselent = 0.001;

double a = -1;

double b = 1;

double[][] x\_var = new double[2][];

do

{

opt\_lyambda = Math.Round((a + b) / 2,4);

l1 = Math.Round(opt\_lyambda - (epselent / 2),4);

l2 = Math.Round(opt\_lyambda + (epselent / 2),4);

x\_var[0] = new double[] { var[0] + l1 \* d\_mas[0], var[1] + l1 \* d\_mas[1] };

x\_var[1] = new double[] { var[0] + l2 \* d\_mas[0], var[1] + l2 \* d\_mas[1] };

f1 = f(x\_var[0]);

f2 = f(x\_var[1]);

if (f1 > f2)

{

a = opt\_lyambda;

}

if (f1 < f2)

{

b = opt\_lyambda;

}

} while (Math.Abs(b - a) > epselent);

opt\_lyambda = (a + b) / 2;

return opt\_lyambda;

}

double norma(double[] var)

{

double max = double.MinValue;

for (int i = 0; i < var.Length; i++)

{

if (var[i] > max)

{

max = var[i];

}

}

return max;

}

//Вспомогательный шаг

int k = 0; int j = 0;

double[][] d\_inner = new double[2][];

d\_inner[0] = new double[] { 1, 0 };

d\_inner[1] = new double[] { 0, 1 };

double[][] x = new double[100][];

int n = 2;

double[][] lyambda = new double[100][];

double[] d\_opt = new double[100];

x[k] = new double[] { x1, x2 };

double[][] y = new double[n + 1][];

y[j] = new double[x[k].Length];

y[j] = x[k];

double eps = epsel;

double lyambda\_optim;

solutionBox.Text = "Выберем в качестве направлений:";

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

for (int i = 0; i < d\_inner.Length; i++)

{

solutionBox.Text += Environment.NewLine+"d" +(i+1)+"=(";

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

for (int ii = 0; ii < d\_inner[i].Length; ii++)

{

solutionBox.Text += $"{ d\_inner[i][ii]} ";

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

}

solutionBox.Text += ")";

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

}

solutionBox.Text += Environment.NewLine + "Критерий для остановки алгоритма ε = " + eps + Environment.NewLine;

do

{

k++;

solutionBox.Text += Environment.NewLine+"Итерация №" +k+" (k)"+Environment.NewLine;

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

solutionBox.Text += Environment.NewLine + "y1 = (";

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

for (int i = 0; i < y[j].Length; i++)

{

solutionBox.Text += Math.Round(y[j][i], 4) + " ";

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

}

solutionBox.Text += ") " + Environment.NewLine;

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

//Шаг1

do

{

j++;

solutionBox.Text += Environment.NewLine + "Итерация №" + j + " (j)" + Environment.NewLine;

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

solutionBox.Text += "Произведем оптимизационное решение задачи минимизации "+ Environment.NewLine;

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

lyambda[j - 1] = new double[] { func\_minimization(y[j - 1], d\_inner[j - 1]) };

solutionBox.Text += "λопт("+j+") = "+ lyambda[j-1][0] + Environment.NewLine;

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

y[j] = new double[y[j - 1].Length];

y[j] = addVecToVec(y[j - 1], multVector(d\_inner[j - 1], lyambda[j - 1][0]));

solutionBox.Text += Environment.NewLine+"y["+(j+1)+"] = (";

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

for (int i = 0;i<y[j].Length;i++)

{

solutionBox.Text += Math.Round(y[j][i],4) + " ";

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

}

solutionBox.Text += ") "+Environment.NewLine;

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

} while (j < n);

x[k] = new double[y[n].Length];

solutionBox.Text += Environment.NewLine+"x["+k+"] = y[" + (n + 1) + "]" + Environment.NewLine;

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

for (int i = 0; i < x[k].Length; i++) x[k][i] = y[n][i];

//Шаг2

Array.Resize<double>(ref d\_opt, 0);

d\_opt = subtractVecToVec(x[k], x[k - 1]);

lyambda\_optim = func\_minimization(x[k], d\_opt);

if (norma(subtractVecToVec(x[k], x[k - 1])) > eps)

{

solutionBox.Text += Environment.NewLine + "||x[k+1] - x[k]|| ≥ ε" + Environment.NewLine;

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

solutionBox.Text += Environment.NewLine + "d=x[" + (k + 1) + "] - x[" + k + "] = (";

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

for (int i = 0; i < d\_opt.Length; i++)

{

solutionBox.Text += Math.Round(d\_opt[i], 4) + " ";

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

}

solutionBox.Text += ")" + Environment.NewLine;

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

solutionBox.Text += Environment.NewLine + "Оптимизационное решение задачи задачи минимизации:" + Environment.NewLine + "λ = " + lyambda\_optim + Environment.NewLine;

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

}

Array.Resize(ref y, 100);

for (int i = 0; i < y.Length; i++)

{

Array.Resize(ref y[i], 0);

}

Array.Resize(ref lyambda, 100);

for (int i = 0; i < lyambda.Length; i++)

{

Array.Resize(ref lyambda[i], 0);

}

j = 0;

y[j] = new double[x[k].Length];

y[j] = addVecToVec(x[k], multVector(d\_opt, lyambda\_optim));

} while (norma(subtractVecToVec(x[k], x[k - 1])) > eps);

solutionBox.Text += Environment.NewLine+Environment.NewLine+"Оптимизация завершена!!" +Environment.NewLine+"Вектор x\* = (";

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

for(int i = 0; i < x[k].Length; i++)

{

solutionBox.Text += Math.Round(x[k][i],4)+" ";

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

}

solutionBox.Text += ")" + Environment.NewLine + "f(x\*) = " + Math.Round(f(x[k]), 4) + Environment.NewLine;

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

}

Результат оптимізації:

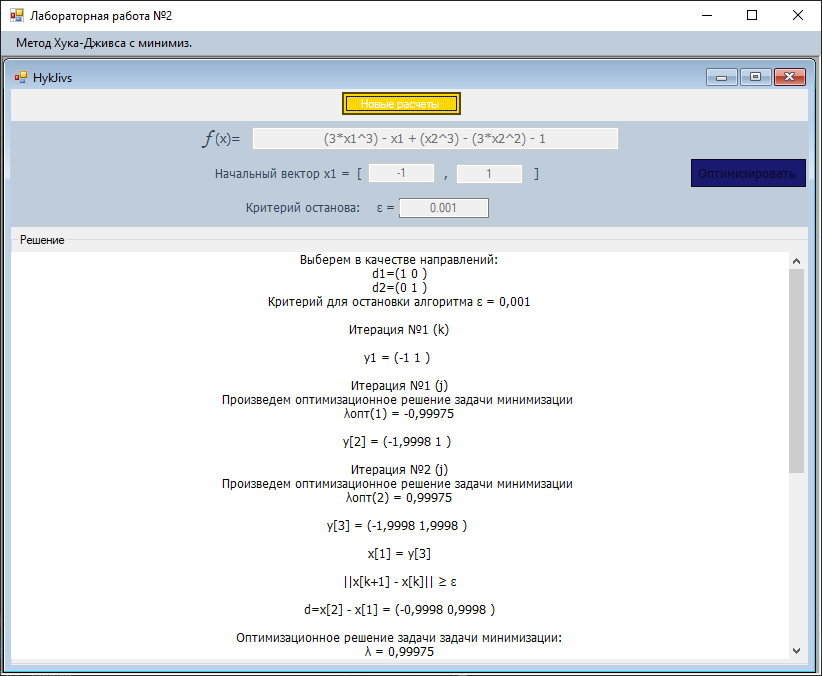


Рисунок 7 – Результати оптимізації

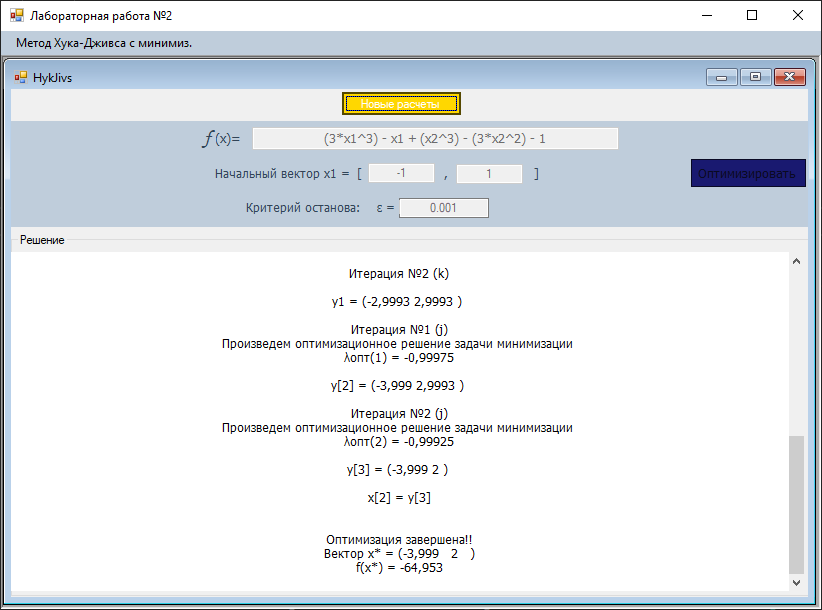


Рисунок 8 – Результати оптимізації

Висновки:

Виконуючи лабораторну роботу отримані навички у сфері безумовної оптимізації методом Хука-Дживса з одномірною мінімізацією. Отримані навички були закріплено за допомогою програмування алгоритмів мовою програмування C#. Закріплено навички у складанні блок-схем алгоритмів.