МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

“ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГИЙ УПРАВЛІННЯ

Лабораторна робота № 1

з курсу «Дослідження операцій»

«Вивчення методів одномірної оптимізації»

Виконав:

студент групи КН 36-а

Кулик Володимир

Перевірив:

Гужва В. О.

ХАРКІВ 2018

Мета роботи:

Вивчити методи одномірної оптимізації , скласти блок-схему алгоритму методу, скласти програму зі складеної блок-схеми. Необхідно мінімізувати функцію методом золотого перетину.

Алгоритм методу золотого перетину:

Початковий етап.

Вибираємо кінцеву дожину інтервалу невизначеності ℇ>0. Обчислюємо

та

, де α=0,618. Вважаємо що k=1.

Основний етап.

Крок 1. Якщо , то оптимальна точка належить інтервалу []. Обчислити та , припинити обчислення. В іншому випадку, якщо ,, то йти до кроку 2, а якщо , то до кроку 3.

Крок 2. Покласти

. Обчислити та . Перейти до кроку 4.

Крок 3. Покласти

. Обчислити та . Перейти до кроку 4.

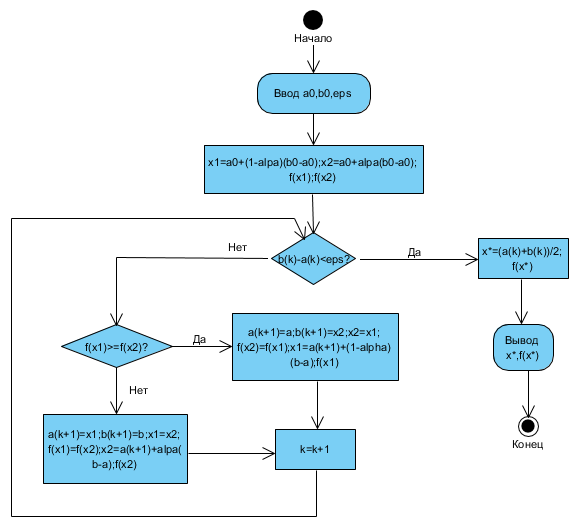
Крок 4. Замінити k на k+1 та перейти до кроку 1.

Зауваження:

При першій реалізації кроку 5 точка може знаходитися за точкою . Для виключення великого екстраполяціонного переміщення слід провести після кроку 5 додаткову перевірку і в разі, коли знаходиться дуже далеко від , замінити точкою, що вирахувана з урахуванням заздалегідь встановленої довжини кроку.

Блок-схема алгоритму:

Блок-схема методу золотого перетину:



Програма:

Метод золотого перетину:

double func\_res(double x)

{

double res=0;

if ((x - 2) < 0)

{

res += Math.Pow(Math.Pow(x - 1, 2), (1.0 / 3.0)) / (Math.Pow(x, 2) + 1); }

if ((x - 2) >= 0)

{

res = Math.Pow(Math.Pow(x - 1, 2), (1.0 / 3.0)) / (Math.Pow(x, 2) + 1);

}

return res;

}

double find(double st, double end, double ep)

{

x1 = Math.Round(st + (1 - alpha) \* (end - st),4);

x2 = Math.Round(st + alpha \* (end - st),4);

double f1 = Math.Round(func\_res(x1),4);

double f2 = Math.Round(func\_res(x2),4);

textBox1.Text = "Шаг 1."+ Environment.NewLine+"x1 = "+ x1+ ", x2="+ x2+ Environment.NewLine+ "f(x1)=" +f1+ ", f(x2)="+f2;

textBox1.SelectionStart = textBox1.TextLength;

textBox1.ScrollToCaret();

do

{

if (f1 <= f2)

{

end = Math.Round(x2,4);

x2 = Math.Round(x1,4);

f2 = Math.Round(f1,4);

x1 = Math.Round(st + (1 - alpha) \* (end - st),4);

f1 = Math.Round(func\_res(x1),4);

count++;

textBox1.Text += Environment.NewLine + Environment.NewLine + "Шаг " + count + "." + Environment.NewLine + "f(x1)>=f(x2) =>" + Environment.NewLine + "Новые значения:" + Environment.NewLine + "b=" + end + ", a=" + st + Environment.NewLine + "x1=" + x1 + ", x2=" + x2 + Environment.NewLine + "f(x1)=" + f1 + ", f(x2)=" + f2;

textBox1.SelectionStart = textBox1.TextLength;

textBox1.ScrollToCaret();

}

if (f1 > f2)

{

st = Math.Round(x1,4);

x1 = Math.Round(x2,4);

f1 = Math.Round(f2,4);

x2 = Math.Round(st + alpha \* (end - st),4);

f2 = Math.Round(func\_res(x2),4);

count++;

textBox1.Text += Environment.NewLine + Environment.NewLine + "Шаг " + count + "." + Environment.NewLine + "f(x1)<f(x2) =>" + Environment.NewLine + "Новые значения:" + Environment.NewLine + "b=" + end + ", a=" + st + Environment.NewLine + "x1=" + x1 + ", x2=" + x2 + Environment.NewLine + "f(x1)=" + f1 + ", f(x2)=" + f2;

textBox1.SelectionStart = textBox1.TextLength;

textBox1.ScrollToCaret();

}

}while(Math.Abs(end - st) >= eps);

textBox1.Text += Environment.NewLine + Environment.NewLine + "b-a<eps" + Environment.NewLine + Math.Round(end,4) + "- (" + Math.Round(st,4) + ") <" + eps;

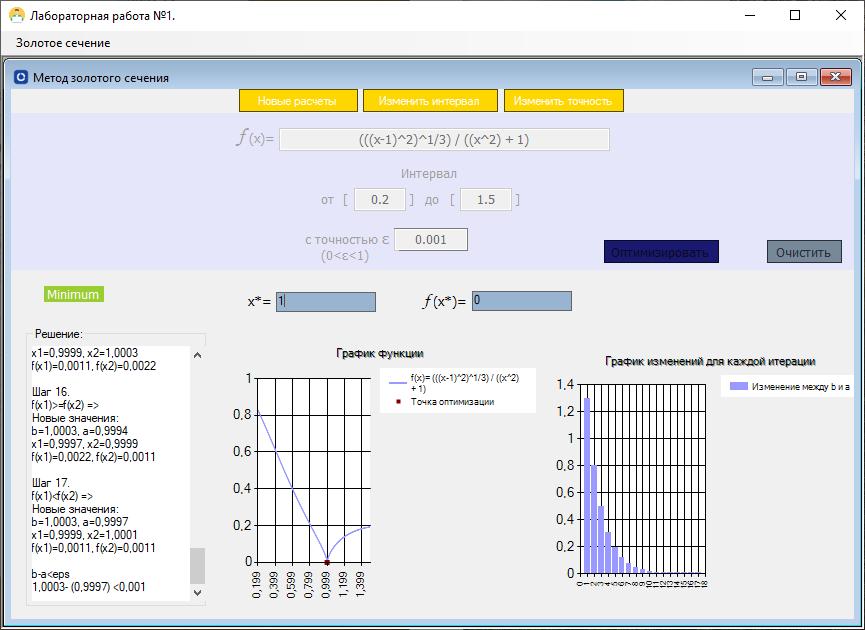
textBox1.SelectionStart = textBox1.TextLength;

textBox1.ScrollToCaret();

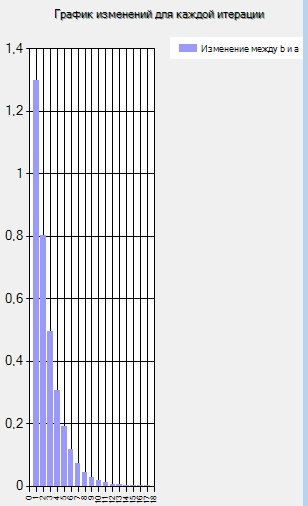
return (st+end)/2.0;

Результати оптимізації:

Метод золотого перетину:



Результат изменений между a и b для каждой итерации:



Висновки:

Виконуючи лабораторну роботу отримані навички у сфері одномірної оптимізації алгоритмом, а саме методом золотого перетину. Отримані навички були закріплені за допомогою програмування алгоритмів мовою програмування C#. Закріплено навички у складанні блок-схем алгоритмів.