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»

Виконала:

студентка групи КН 36-б

Ликова Маргарита

Перевірив:

Гужва В. О.

ХАРКІВ 2018

Мета роботи:

Вивчити методи одномірної оптимізації , скласти блок-схеми алгоритмів вибраних методів, скласти програми зі складених блок-схем. Необхідно мінімізувати функцію двома методами: методом золотого перетину та методом квадратичної апроксимації. Порівняти отримані результати.

Алгоритми методів:

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.

Алгоритм методу квадратичної апроксимації.

Нехай початкова точка, обрана величина кроку по осі .

Крок 1. Обчислити .

Крок 2. Обчислити і .

Крок 3. Якщо > : ,

якщо ≤ : .

Крок 4. Обчислити та знайти , та точку яка відповідає цьому значенню: .

Крок 5. По трьом точкам обчислити , використовуючи формулу оцінювання за допомогою квадратичної апроксимації:

Необхідно обчислити таким чином, що значення квадратичної функції співпадали зі значеннями функції в трьох точках.

У точці : .

У точці : .

У точці : .

Оскільки унімодальна, то можна очікувати, що виявиться величина прийнятна для оцінки координати .

Крок 6. Перевірка на закінчення пошуку:

1. Чи є різниця достатньо малою?
2. Чи є різниця достатньо малою?

Якщо ці дві умови виконуються: зупинитись.

В інакшому випадку перейти до кроку 7.

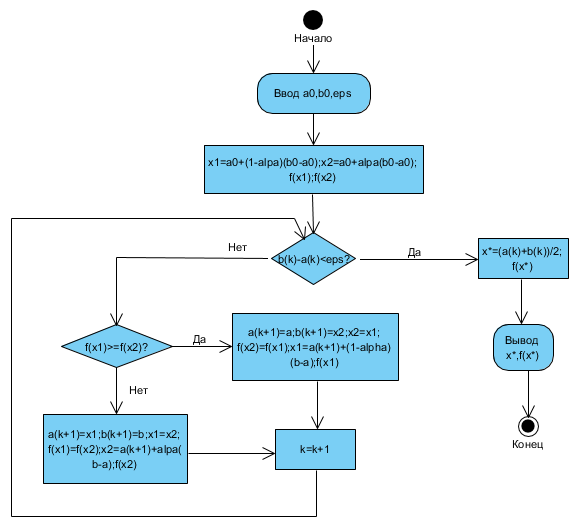
Крок 7. Вибрати найкращу точку () та дві точки по обидва боки від неї. Позначити ці точки в природному порядку та перейти до кроку 4.

Зауваження:

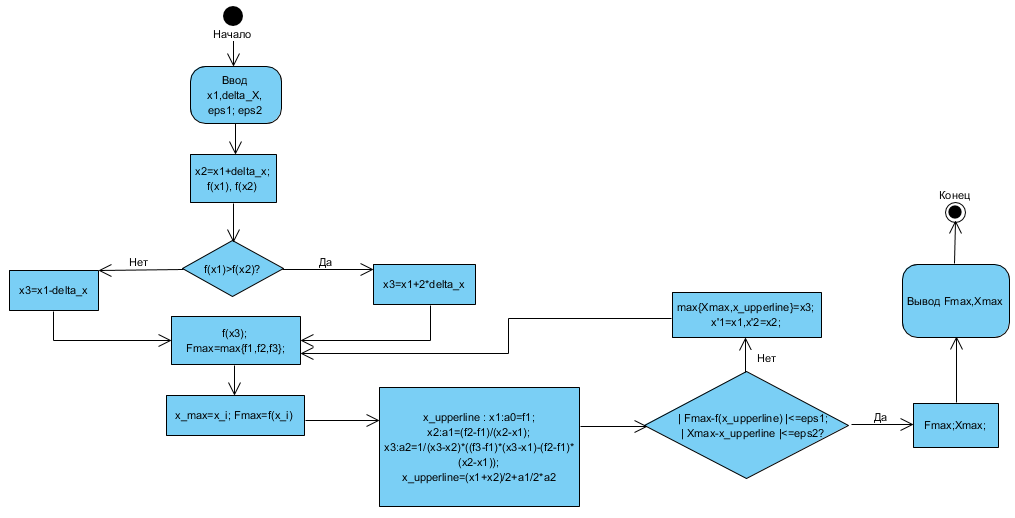
При першій реалізації кроку 5 точка може знаходитися за точкою . Для виключення великого екстраполяціонного переміщення слід провести після кроку 5 додаткову перевірку і в разі, коли знаходиться дуже далеко від , замінити точкою, що вирахувана з урахуванням заздалегідь встановленої довжини кроку.

Блок-схеми алгоритмів:

Блок-схема методу золотого перетину:



Блок-схема методу квадратичної апроксимації:



Програми:

Метод золотого перетину:

double func\_res(double x)

{

double res=0;

if ((x - 2) < 0)

{

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

}

if ((x - 2) >= 0)

{

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

}

//res = Math.Pow(x, 4)-14\* Math.Pow(x, 3)+60\* Math.Pow(x, 2)-70\*x;

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;

}

Метод квадратичної апроксимації:

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

double delta = double.Parse(deltaxBox.Text.Replace(".", ","));

double eps1 = double.Parse(eps1Box.Text.Replace(".", ","));

double eps2 = double.Parse(eps2Box.Text.Replace(".", ","));

double x2, f1, f2, f3, fMin, a0, a1, a2, x\_upperline, newx1, newx2;

double xMin = 0.0,x3 = 0;

int it\_count = 1;

double func(double x)

{

double res;

if (x - 2 < 0)

{

res = Math.Round(Math.Pow(x, 2) \* (x + 1) \* Math.Pow(-1 \* (x - 2), (1.0 / 3.0)),4);

}

else

{

res = Math.Round(Math.Pow(x, 2) \* (x + 1) \* Math.Pow((x - 2), (1.0 / 3.0)),4);

}

return res;

}

x2 = Math.Round(x1 + delta,4);

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

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

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

conditionBox.SelectionStart = conditionBox.TextLength;

conditionBox.ScrollToCaret();

if (f1 > f2)

{

x3 = Math.Round(x1 + 2 \* delta,4);

conditionBox.Text += Environment.NewLine+"f(x1)>f(x2) => x3=" + x3;

conditionBox.SelectionStart = conditionBox.TextLength;

conditionBox.ScrollToCaret();

}

else if (f1 <= f2)

{

x3 = Math.Round(x1 - delta,4);

conditionBox.Text += Environment.NewLine + "f(x1)<=f(x2) => x3=" + x3;

conditionBox.SelectionStart = conditionBox.TextLength;

conditionBox.ScrollToCaret();

}

f3 = Math.Round(func(x3),4);

solutionBox.Text += Environment.NewLine + "f(x3)=" + f3;

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

fMin = Math.Min(Math.Min(f1, f2), f3);

if (fMin == f1) xMin = x1;

if (fMin == f2) xMin = x2;

if (fMin == f3) xMin = x3;

solutionBox.Text += Environment.NewLine + "x(min)=" + xMin + ", F(x(min))=" + fMin + Environment.NewLine;

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

a0 = Math.Round(f1,4);

a1 = Math.Round((f2 - f1) / (x2 - x1),4);

a2 = Math.Round((1 / (x3 - x2)) \* ((f3 - f1) / (x3 - x1) - (f2 - f1) / (x2 - x1)),4);

x\_upperline = Math.Round(((x1 + x2) / 2) + (a1 / 2 \* a2),4);

solutionBox.Text += Environment.NewLine + Environment.NewLine + "a0=" + a0 + ", a1=" + a1 + ", a2=" + a2 + "=>" + Environment.NewLine + "x(upperline)=" + x\_upperline;

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

if (Math.Abs(fMin - func(x\_upperline)) > eps1 && Math.Abs(xMin - x\_upperline) > eps2)

{

do

{

if(xMin<0&&x\_upperline<0) x3 = Math.Round(Math.Max(xMin, x\_upperline), 4);

else if (xMin > 0 && x\_upperline > 0) x3 = Math.Round(Math.Min(xMin, x\_upperline),4);

solutionBox.Text += Environment.NewLine + Environment.NewLine + "|F(x(min))-f(x(upperline))|> ε1," + Environment.NewLine + "|x(min)-x(upperline)|> ε2 =>" + Environment.NewLine + "x3=min{x(min),x(upperline)}=" + x3;

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

newx1 = Math.Round(x3 - delta,4);

newx2 = Math.Round(x3 + delta+(delta/2),4);

x1 = newx1; x2 = newx2;

it\_count++;

solutionBox.Text += Environment.NewLine + "Новые значения для x1 и x2:" + Environment.NewLine + "x1=" + x1 + ", x2=" + x2 + Environment.NewLine + "Переходим на новую итерацию №" + it\_count;

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

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

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

f3 = Math.Round(func(x3),4);

solutionBox.Text += Environment.NewLine + "f(x1)=" + f1 + ", f(x2)=" + f2 + ", f(x3)=" + f3;

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

fMin = Math.Min(Math.Min(f1, f2), f3);

if (fMin == f1) xMin = x1;

if (fMin == f2) xMin = x2;

if (fMin == f3) xMin = x3;

solutionBox.Text += Environment.NewLine+ Environment.NewLine + "x(min)=" + xMin + ", F(x(min))=" + fMin + Environment.NewLine;

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

a0 = Math.Round(f1,4);

a1 = Math.Round((f2 - f1) / (x2 - x1),4);

a2 = Math.Round((1 / (x3 - x2)) \* (((f3 - f1) / (x3 - x1)) - a1),4);

x\_upperline = Math.Round(((x1 + x2) / 2) + (a1 / 2 \* a2),4);

solutionBox.Text += Environment.NewLine + Environment.NewLine + "a0=" + a0 + ", a1=" + a1 + ", a2=" + a2 + "=>" + Environment.NewLine + "x(upperline)=" + x\_upperline;

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

if (x\_upperline-x3>=delta \* 3)

{

do

{

solutionBox.Text += Environment.NewLine + Environment.NewLine + "x\_upperline находится дальше,чем в три раза от x3:" + Environment.NewLine + "x(upperline)-x3=" + (x\_upperline - x3) + Environment.NewLine + "Переопределим точки:";

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

x1 = x2;

x2 = x3;

x3 =Math.Round(x2 + delta,4);

it\_count++;

solutionBox.Text += Environment.NewLine + Environment.NewLine + "Новые значения для x1 и x2:" + Environment.NewLine + "x1=" + x1 + ", x2=" + x2 + ", x3=" + x3 + Environment.NewLine + "Переходим на новую итерацию №" + it\_count;

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

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

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

f3 = Math.Round(func(x3),4);

solutionBox.Text += Environment.NewLine +"f(x1)="+f1+ ", f(x2)="+f2 + ", f(x3)=" + f3;

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

fMin = Math.Min(Math.Min(f1, f2), f3);

if (fMin == f1) xMin = x1;

if (fMin == f2) xMin = x2;

if (fMin == f3) xMin = x3;

solutionBox.Text += Environment.NewLine + "x(min)=" + xMin + ", F(x(min))=" + fMin + Environment.NewLine;

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

a0 = Math.Round(f1,4);

a1 = Math.Round((f2 - f1) / (x2 - x1),4);

a2 = Math.Round((1 / (x3 - x2)) \* ((f3 - f1) / (x3 - x1) - a1),4);

x\_upperline = ((x1 + x2) / 2) + (a1 / 2 \* a2);

solutionBox.Text += Environment.NewLine + Environment.NewLine + "a0=" + a0 + ", a1=" + a1 + ", a2=" + a2 + "=>" + Environment.NewLine + "x(upperline)=" + x\_upperline;

solutionBox.SelectionStart = solutionBox.TextLength;

solutionBox.ScrollToCaret();

} while (x\_upperline - x3 >= delta \* 3);

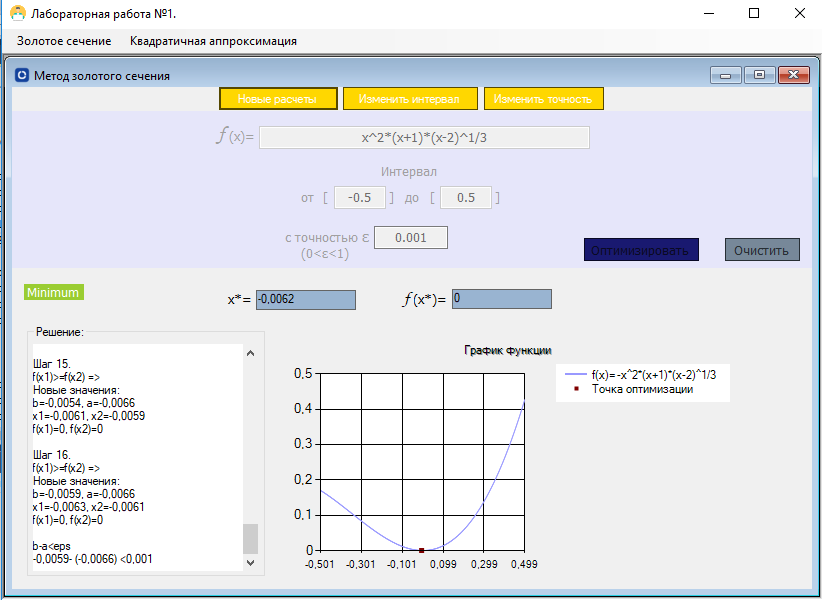
}

} while (Math.Abs(fMin - func(x\_upperline)) > eps1 && Math.Abs(xMin - x\_upperline) > eps2 && xMin<=0.5);

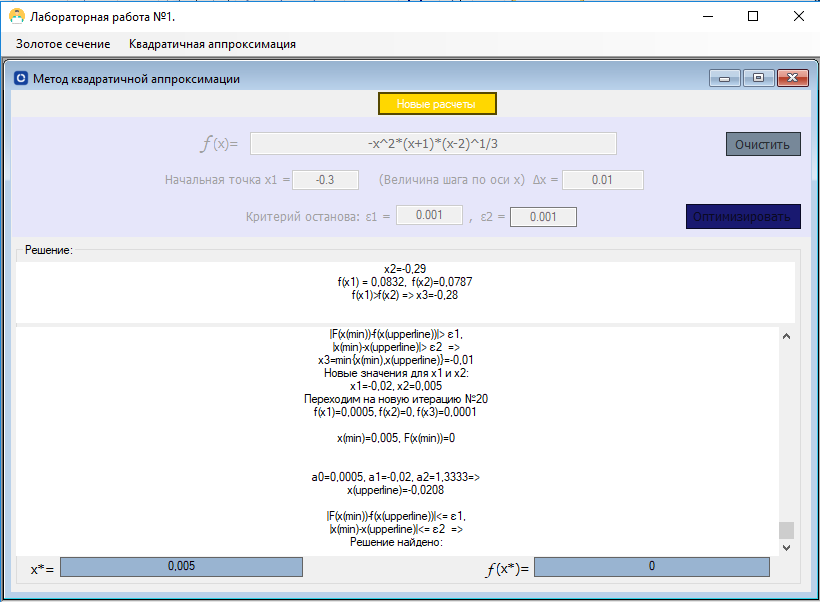
}

Результати оптимізації:

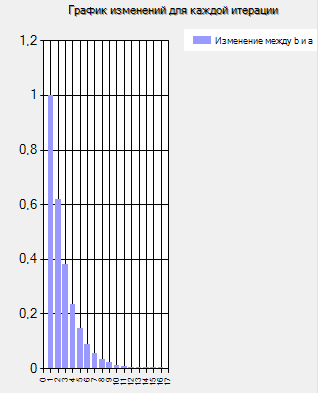
Метод золотого перетину:



Метод квадратичної апроксимації:



Результат изменений между a и b для каждой итерации:



Висновки:

Виконуючи лабораторну роботу отримані навички у сфері одномірної оптимізації алгоритмами,а саме методом золотого перетину і методом квадратичної апроксимації. Отримані навички були закріплено за допомогою програмування алгоритмів мовою програмування c#. Закріплено навички у складанні блок-схем алгоритмів.