МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ СИСТЕМ

ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ И ФИЗИКО-МАТЕМАТИЧЕСКИХ НАУК

КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Методы поиска условного экстремума

Отчет по лабораторной работе №3

02.03.02 Фундаментальная информатика   
и информационные технологии

Выполнил: студент 3 курса \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Савин Д.А.

Проверил: к. б. н., доцент \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Чупраков П. Г.

Киров

2020 г.

**Цель лабораторной работы:**

Целью данной лабораторной работы является изучение методов поиска условного экстремума, применение этих методов на практике (реализация методов для различных функций в какой-либо среде программирования, а также проверка при помощи математических пакетов) и сравнение методов по их эффективности.

**Краткое описание алгоритмов**

Метод штрафов:

1. На вход получить начальную точку вне области допустимых решений, начальное значение штрафа и число для увеличения штрафа, погрешность.
2. Составить вспомогательную функцию с учетом условий. Найти минимум вспомогательной функции.
3. Найти значение штрафной функции от найденной точки минимума вспомогательной функции.
4. Если значение функции штрафа меньше погрешности, то точка минимума – искомая точка минимума функции, принять найденную точку как текущую, увеличить значение штрафа на заданное число и вернуться на 2 шаг.

**Листинг программы с комментариями**

Основная программа:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Meths

{

class Program

{

static double x0, y0;

static double[] x = { x0, y0 };

static double l = 0.01;

static double Func;

static double limitationInc(double x1, double x2)//Ограничения

{

if (limitation(x1, x2) > 0)

return limitation(x1, x2);

else return 0;

}

static double limitation(double x1, double x2)

{

return -x2 + 2;

}

static double fun(double x1, double x2)//Функция

{

return 100 \* Math.Pow(x2 - x1 \* x1, 2) + Math.Pow(1 - x1, 2);

}

public static double f(double[] x)

{

return 100 \* Math.Pow(x[1] - x[0] \* x[0], 2) + Math.Pow(1 - x[0], 2);

}

static double fine(double x1, double x2)//Штрафная функция

{

double GSum = 0, GPlusSum = 0;

GPlusSum += limitationInc(x1, x2) \* limitationInc(x1, x2);

return 0.5 \* l \* (GSum + GPlusSum);

}

static void Min(double[] x)//минимум

{

int n = x.Length;

double y = 2;

int k = 0;

double f0 = F(x), e = 0.001;

k++;

while (y > e \* e)

{

k++;

double f1 = F(x), f2 = 0, f3 = 0;

for (int i = 0; i < n; i++)

{

x[i] += +y;

f2 = F(x);

if (f2 > f1)

{

x[i] -= y \* 2;

f3 = F(x);

if (f3 > f1 || f2 == f3)

{

x[i] += y;

}

}

}

if (f0 == f1)

{

y = y / 2;

k--;

}

f0 = f1;

}

x0 = x[0];

y0 = x[1];

}

static double fine(double[] x)

{

double sumI = 0, sumII = 0;

sumII += limitationInc(x[0], x[1]) \* limitationInc(x[0], x[1]);

return 0.5 \* l \* (sumI + sumII);

}

static double F(double x1, double x2)

{

return fun(x1, x2) + fine(x1, x2);

}

static double F(double[] x)

{

return f(x) + fine(x);

}

static void Main(string[] args)

{

Console.WriteLine("Начальная координата: ");

x0 = double.Parse(Console.ReadLine());

Console.WriteLine("Конечная координата: ");

y0 = double.Parse(Console.ReadLine());

Console.WriteLine("Увеличение штрафа: ");

double fineinc = double.Parse(Console.ReadLine());

Console.WriteLine("Начальный штраф: ");

l = double.Parse(Console.ReadLine());

Console.WriteLine("Погрешность: ");

double eps = double.Parse(Console.ReadLine());

int k = 0;

s2:

Func = F(x0, y0);

double[] arg = { x0, y0 };

double PFunc = fine(x0, y0);

if (PFunc <= eps)

{

Console.WriteLine("x1 = {0}\nx2 = {1}\nf(x1,x2) = {2}\nk={3}\n", x0.ToString(), y0.ToString(), fun(x0, y0), k);

}

else

{

Console.WriteLine("x1 = {0}\nx2 = {1}\nf(x1,x2) = {2}\n", x0.ToString(), y0.ToString(), fun(x0, y0));

Min(arg);

l \*= fineinc;

k++;

goto s2;

}

Console.WriteLine();

}

}

}

**Результаты выполнения программы**

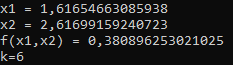
Для всех функций параметры:

* Начальный штраф = 0.01
* Увеличение штрафа = 10
* Погрешность = 0.01

Первая функция:

100\*(x1 - x0 \* x0)2 + (1 - x0)2 , x0-x1<=-1

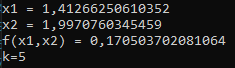
x0=1 x1=0



Вторая функция:

100\*(x1 - x0 \* x0)2 + (1 - x0)2 , x1>=2

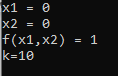
x0=1 x1=0



Третья функция:

(x1 - x0 \* x0)2 + (1 - x0)2 , x1<=0, x2<=0

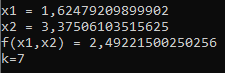
x0=1 x1=1



Четвёртая функция:

(x1 - x0 \* x0)2 + 5\*(1 - x0)2, x0+x1>=5

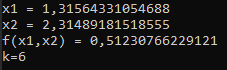
x0=0 x1=0



Пятая функция:

10\*(x1 - x0 \* x0\*x0)2 + (1 - x0)2, x0-x1<=-1

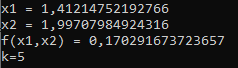
x0=2 x1=0



Шестая функция:

10\*(x1 - x0 \* x0)2 + (1 - x0)2, x1>=2

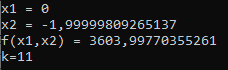
x0=0 x1=0



Седьмая функция:

(x0-x1)2 + 100\*(x0+x1-4)2, -0.1\*x0\*x0+x1<=0, x1<=-2, x0<0

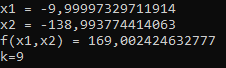
x0=0 x1=0



Восьмая функция:

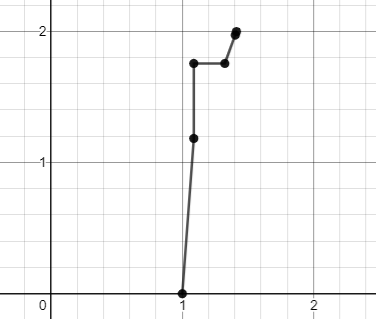
(3-x0)2 + 100\*(x1+(x0-2)^2-5)2, 9\*(x0-2)2+25x12>=500, x0<=-10, x22>17

x0=0 x1=0



**Графическая интерпретация для двумерного случая**

x=x1, а y=x2



**Проверка вычислений в математическом пакете**

Первой функция:



Второй функция:



Третья функция:



Четвертая функция:



Пятая функция:



Шестая функция:



Седьмая функция:



Восьмая функция:

