**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**

БАЛТИЙСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ВОЕНМЕХ» им. Д.Ф. УСТИНОВА



Дисциплина МЕТОДЫ ОПТИМИЗАЦИИ

практическая работа № 1

Метод наискорейшего спуска.

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил студент | | Анкудинов А.Н. | |
| Фамилия И.О. | | | |
| Группа | А571 | |  |
| Преподаватель | |  | |
|  | | Фамилия И. О. | |

Санкт-Петербург

2020 г.

*Цель работы* – минимизировать функцию, используя метод наискорейшего спуска.

*Исходные данные*

*Вариант 1*

,

– начальное значение, – точность вычисления.

*Описание метода наискорейшего спуска*

Суть данного метода в том, что на каждой итерации величина шага выбирается из условия:

.

*Ход работы*

Сначала необходимо вычислить значение функции в начальной точке и её градиент.

В начале первой итерации вычисляется градиент в точке . Тогда

Величину шага можно найти из условия (1) методом золотого сечения, подставив найденные величины и далее вычислить . С помощью полученного шага, из (2) найдём и вычислим значение функции в этой точке

Критерий остановки: (3).

С каждой итерацией приближается к своему минимальному значению.

*Результаты*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *i* |  |  |  |  |
| 1 | -0.092756 | 0.0713489 | 0.197585 | 1.02128 |
| 2 | 0.00229071 | 0.00572429 | 0.317166 | 1.00009 |

*Вывод:* методом наискорейшего спуска с заданной точностью было найдено минимальное значение исходной функции .

*Код программы*

#include <iostream>

using namespace std;

double **F**(const double\* x) //для метода наискорейшего спуска

{

return 2\*x[0]\*x[0] + x[0]\*x[1] + 2.1\*x[1]\*x[1] + 1;

}

double **F**(const double\* x, const double\* grad\_x, const double h) //для метода зол. сечения

{

double x1 = x[0] - h\*grad\_x[0];

double x2 = x[1] - h\*grad\_x[1];

return 2\*x1\*x1 + x1\*x2 + 2.1\*x2\*x2 + 1;

}

int **main**()

{

double x[] = {0.5, 1};

double F\_old = F(x);

const double ALPHA = 0.618;

const double e = 0.01; //точность внутреннего цикла - метода зол. сечения

const double eps = 0.05; //точность внешнего цикла - наискорейшего спуска

double grad\_x[2], a, b, l, mu, hmin, F\_new;

int i = 0;

bool firsttime = true;

do

{

++i;

if(!firsttime)

F\_old = F\_new;

grad\_x[0] = 4\*x[0] + x[1];

grad\_x[1] = x[0] + 4.2\*x[1];

a = 0;

b = 1;

l = a + (1-ALPHA)\*(b-a);

mu = a + ALPHA\*(b-a);

while(b - a > e) //внутренний цикл - метод зол. сечения

{

if(F(x, grad\_x, l) > F(x, grad\_x, mu))

{

a = l;

l = mu;

mu = a + ALPHA\*(b-a);

}

else

{

b = mu;

mu = l;

l = a + (1-ALPHA)\*(b-a);

}

}

hmin = a + (b-a)/2;

x[0] = x[0] - hmin\*grad\_x[0];

x[1] = x[1] - hmin\*grad\_x[1];

F\_new = F(x);

firsttime = false;

cout << "i = " << i << "\tx1 = " << x[0] << "\tx2 = " << x[1] << "\th = " << hmin << "\tF(x) = " << F\_new << endl;

} while(F\_old - F\_new > eps); //внешний цикл - метод наискорейшего спуска

return 0;

}