Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ

Кафедра автоматизированных систем (АСУ)

**Оптимизация функции одной переменной**

Отчет по лабораторной работе № 1 по дисциплине

«Методы оптимизации»

|  |
| --- |
| Выполнила: |
| Студенты гр. 439-1 |
| Зозуля Е.Д.  Рахматуллин Т.Т. |
| \_\_.\_\_.2021 г. |
|  |

|  |
| --- |
| Руководитель: |
| А.А. Шелестов |
| « »\_\_\_\_\_\_\_\_\_\_2021 г. |

2021

**Задание**

Найти минимум функции одной переменной, используя два прямых метода из трех (метод равномерного поиска, метод деления отрезка пополам, метод золотого сечения).

Точность 

**Вариант №9**



График функции по варианту показан на Рисунке 1:

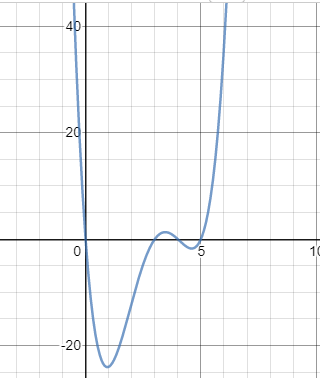


Рисунок 1 – График функции

**Метод равномерного поиска**

Суть метода: интервал [a0, b0] делится на N + 1 равных подинтервалов, в  
каждой точке (границах подинтервалов) вычисляется значение функции.  
Выбирается точка, в которой значение функции минимально.

**Алгоритм:**

Шаг 1. Задать начальный интервал неопределенности L0 = [a0, b0], N - количество вычислений функции;

Шаг 2. Вычислить точки , i = 1...N, равноотстоящие друг от друга;

Шаг 3. Вычислить значения функции в N найденных точках: f(xi), i = 1...N;

Шаг 4. Среди точек xi, i = 1...N, найти такую, в которой функция принимает наименьшее значение: f(xk) = min f(xi).

**Метод дихотомии**

Суть метода: вычисляется середина интервала [a0, b0] и две точки по обе стороны от этой середины, и рассчитывается значение функции в этих точках. Если значение функции в левой точке меньше, чем значение функции в правой точке, значит, функция возрастает на этом промежутке (рис.1.3), и происходит изменение правой границы. Иначе – функция убывает и происходит изменение левой границы.

**Алгоритм:**

Шаг 1. Задать исходные данные: начальный интервал неопределённости L0 = [a0, b0], ε> 0 – малое число, l> 0 – точность (ε ∈ (0.2l));

Шаг 2. Положить k = 0;

Шаг 3. Вычислить , , , ;

Шаг 4. Сравнить с :

а) если ≤, положить ak+1 = ak, bk+1 = zk и перейти к шагу 5;

б) если >, положить ak+1 = yk, bk+1 = bk.

Шаг 5. Вычислить Lk = |bk+1 – ak+1| и проверить условие окончания:

а) если Lk ≤ l, процесс поиска завершается и в качестве приближённого решения можно взять середину последнего интервала: ;

б) если Lk > l, положить k = k + 1 и перейти к шагу 3.

**Листинг вычислительных алгоритмов**

#include <iostream>

#include <math.h>

#define M\_PI 3.14159265358979323846

using namespace std;

double Func(double x) {

return x\*x\*x\*x - 12\*x\*x\*x + 47\*x\*x - 60\*x;

}

//равномерный поиск

void EvenSearch()

{

double a = 0.0, x\_min, y\_min;

double b = M\_PI / 4.0;

int N = 100;

double\* x = new double[N];

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

{

x[i] = a + (i \* ((b - a) / (N + 1)));

if (i == 0) {

x\_min = x[i];

y\_min = Func(x[i]);

}

else if (Func(x[i]) <= y\_min) {

y\_min = Func(x[i]);

x\_min = x[i];

}

}

cout << "x\_min= " << x\_min << "\ny\_min= " << y\_min;

}

//дихотомия

void Dichotomy()

{

double a = -5.0, b = 5.0, e = 0.001, l = 0.002, k, y\_k, f\_yk, z\_k, f\_zk, L = 1;

k = 0;

while (L > l)

{

y\_k = (a + b - e) / 2;

f\_yk = Func(y\_k);

z\_k = (a + b + e) / 2;

f\_zk = Func(z\_k);

if (f\_yk <= f\_zk)

b = z\_k;

else

a = y\_k;

L = abs(b - a);

}

cout << "min\_x = " << (a + b) / 2 << "\nmin\_y = " << Func((a + b) / 2);

}

int main()

{

setlocale(0,"rus");

cout << "Равномерный поиск:\n";

EvenSearch();

cout << "\nДихотомия:\n";

Dichotomy();

}

**Результаты работы программы**

