**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ**

**Московский государственный технический университет**

**им. Н.Э. Баумана**

**(МГТУ им. Н.Э. Баумана)**

**Кафедра «Информационная безопасность» (ИУ8)**

Отчёт

Лабораторная работа № 1

По дисциплине: «Теория систем и системный анализ»

# Тема: «Исследование методов прямого поиска экстремума унимодальной

# функции одного переменного»

# Вариант 8

Выполнил: Забродина М.П.,

студент группы ИУ8-32

Проверил: Коннова Н.С.

Доцент каф. ИУ8

г. Москва 2019 г.

**Цель:**

Исследовать функционирование и провести сравнительный анализ различных алгоритмов прямого поиска экстремума (пассивный поиск, метод дихотомии, золотого сечения, Фибоначчи) на примере унимодальной функции одного переменного.

**Условие задачи:**

На интервале [-2;5] задана унимодальная функция одного переменного f(x)=cos(x)\*th(x). Используя метод золотого сечения, найти интервал нахождения минимума f(x) при заданной наибольшей допустимой длине интервала неопределенности Е=0,1. Провести сравнение с методом оптимального пассивного поиска. Результат в зависимости от количества точек разбиения представить в виде таблицы.

**График функции f(x)=cos(x)\*th(x) на интервале [-2;5].**



//Кафедра "Информационная безопасность"

//ИУ8-32

//Забродина М.П.

//Лабораторная работа №1(Теория систем и системный анализ)

#include <iostream>

#include <math.h>

#include <iomanip>

**using** **namespace** std;

**const** **double** goldenRatio = (1 + sqrt(5)) / 2;

**double** func(**double** x) {

**return** cos(x)\*tanh(x);

}

**int** main() {

**double** a = 1.5, b = 4;

**double** e = 0.1;

**double** x1, x2;

cout <<"1. Sequential search (The golden ratio method):" <<endl;

cout <<setfill('-') <<setw(76) <<"-" <<endl;

cout <<"| Start of the | End of the | Length of | | |" <<endl;

cout <<"| interval | interval | the interval | f(ak) | f(bk) |" <<endl;

cout <<"| (ak) | (bk) | (l) | | |" <<endl;

cout <<setfill('-') <<setw(76) <<"-" <<endl;

**while** (fabs(b - a) > e) {

x1 = b - (b - a) / goldenRatio;

x2 = a + (b - a) / goldenRatio;

cout <<setfill(' ') <<"|" <<setw(14) <<fixed <<setprecision(5) <<x1 <<"|" <<setw(14) <<x2 <<"|" <<setw(14) <<abs(b-a) <<"|" <<setw(14) <<func(x1) <<"|" <<setw(14) <<func(x2) <<"|" <<endl;

**if** (func(x1) <= func(x2) ) // Условие для поиска минимума

b = x2;

**else** a = x1;

} // Выполняем, пока не достигнем заданной точности

x1 = b - (b - a) / goldenRatio;

x2 = a + (b - a) / goldenRatio;

cout <<setfill(' ') <<"|" <<setw(14) <<fixed <<setprecision(5) <<x1 <<"|" <<setw(14) <<x2 <<"|" <<setw(14) <<abs(b-a) <<"|";

**if** (b-a < e) cout <<" l < epsilon |" <<endl;

cout <<setfill('-') <<setw(76) <<"-" <<endl;

cout <<"Minimum is reached at x = " <<fixed <<setprecision(3) << (a + b) / 2 <<" +- " <<(b-a)/2 <<endl;

cout <<endl <<endl;

cout <<"2. Optimal passive search:" <<endl;

a = 1.5;

b = 4;

**int** N = 1;

cout <<setfill('-') <<setw(32) <<"-" <<endl;

cout << "| Number of | Value of x |" <<endl;

cout << "| points(N) | in the min |" <<endl;

cout <<setw(32) <<"-" <<endl;

**while** ((b - a) / N > e){

cout <<"|" <<setfill(' ') <<setw(13) <<N <<"|";

**double** a1 = a, b1 = b;

**double** d = (b1 - a1) / (N + 1);

**while** (b1 - a1 > 2 \* d){

x1 = a1 + d;

x2 = b1 - d;

**if** (func(x1) <= func(x2)) {

b1 = x2;

} **else** a1 = x1;

} cout <<" " <<(b1 + a1)/2 <<" +- " <<d <<"|" <<endl;

N++;

}

cout <<setfill('-') <<setw(32) <<"-" <<endl;

**return** 0;

}

**1.3** Выходные данные

1. Sequential search (The golden ratio method):

-------------------------------------------------------------------------------

| Start of the | End of the | Length of | | |

| interval | interval | the interval | f(ak) | f(bk) |

| (ak) | (bk) | (l) | | |

-------------------------------------------------------------------------------

| 2.45492| 3.04508| 2.50000| -0.76203| -0.99085|

| 3.04508| 3.40983| 1.54508| -0.99085| -0.96214|

| 2.81966| 3.04508| 0.95492| -0.94190| -0.99085|

| 3.04508| 3.18441| 0.59017| -0.99085| -0.99566|

| 3.18441| 3.27051| 0.36475| -0.99566| -0.98884|

| 3.13119| 3.18441| 0.22542| -0.99614| -0.99566|

| 3.09830| 3.13119| 0.13932| -0.99500| -0.99614|

| 3.13119| 3.15152| 0.08610| l < epsilon |

--------------------------------------------------------------------------------

Minimum is reached at x = 3.141 +- 0.043

2. Optimal passive search:

-------------------------------------

| Number of | Value of x |

| points(N) | in the min |

-------------------------------------

| 1| 2.750 +- 1.250|

| 2| 3.167 +- 0.833|

| 3| 3.375 +- 0.625|

| 4| 3.000 +- 0.500|

| 5| 2.958 +- 0.417|

| 6| 3.286 +- 0.357|

| 7| 3.062 +- 0.312|

| 8| 3.028 +- 0.278|

| 9| 3.250 +- 0.250|

| 10| 3.205 +- 0.227|

| 11| 3.167 +- 0.208|

| 12| 3.231 +- 0.192|

| 13| 3.196 +- 0.179|

| 14| 3.083 +- 0.167|

| 15| 3.219 +- 0.156|

| 16| 3.044 +- 0.147|

| 17| 3.097 +- 0.139|

| 18| 3.211 +- 0.132|

| 19| 3.125 +- 0.125|

| 20| 3.167 +- 0.119|

| 21| 3.205 +- 0.114|

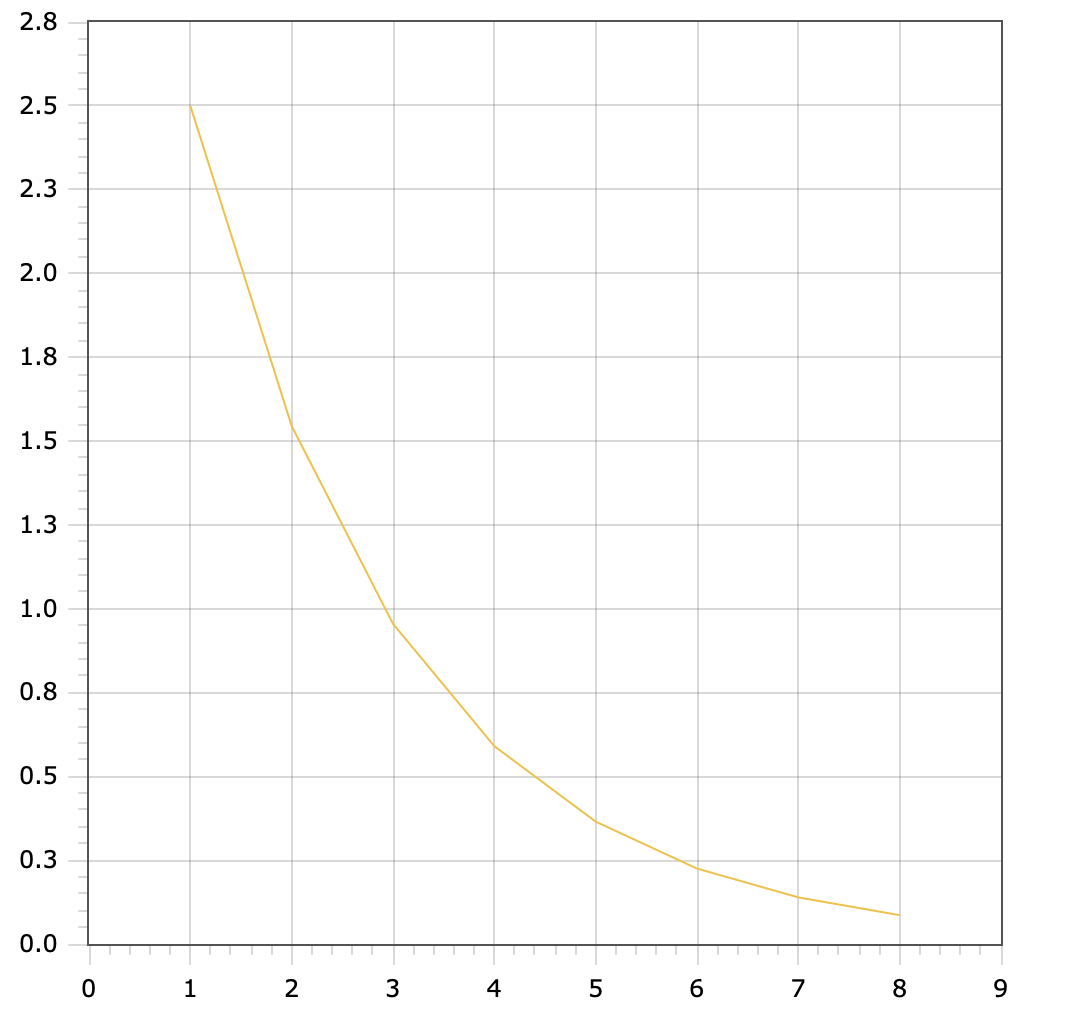
| 22| 3.130 +- 0.109|

| 23| 3.115 +- 0.104|

| 24| 3.100 +- 0.100|

-------------------------------------

**График зависимости погрешности от числа точек N для метода золотого сечения**



**График зависимости погрешности от числа точек N для оптимального пассивного поиска**

****

**Вывод**

Из полученных таблиц и графиков видно, что метод золотого сечения значительно эффективнее метода пассивного поиска при поиске экстремума унимодальной функции одного переменного.