



**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)

КАФЕДРА «Информационная безопасность» (ИУ8)

Отчёт

**по лабораторной работе № 1
по дисциплине «Теория систем и системный анализ»**

**Тема: «Исследование методов прямого поиска экстремума унимодальной функции
одного переменного»**

Вариант 10

**Выполнил:
Минькова А. А., студент
группы ИУ8-31**

**Проверил: Коннова Н.С.,
доцент каф. ИУ8**

**г. Москва,
2020 г.**

1. Цель работы

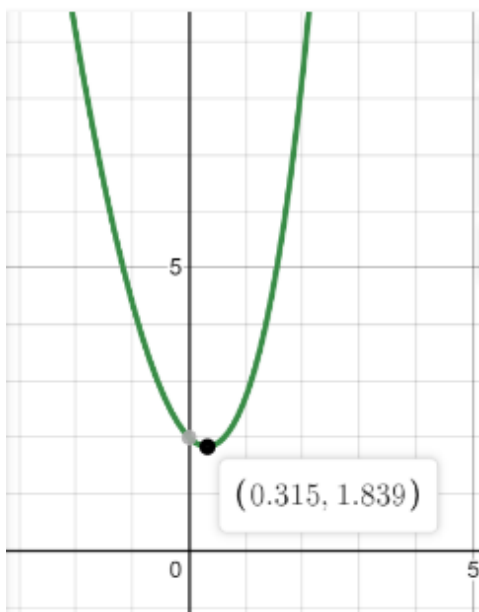
Исследовать функционирование и провести сравнительный анализ различных алгоритмов прямого поиска экстремума (пассивный поиск, метод дихотомии, золотого сечения, Фибоначчи) на примере унимодальной функции одного переменного.

2. Условие задачи

На интервале $[-2,4]$ задана унимодальная функция одного переменного $f(x)=(1-x)^2+\exp(x)$. Используя метод Фибоначчи, найти интервал нахождения минимума $f(x)$ при заданной наибольшей допустимой длине интервала неопределенности $\varepsilon = 0,1$. Провести сравнение с методом оптимального пассивного поиска. Результат, в зависимости от числа точек разбиения N , представить в виде таблицы.

3. Ход работы

Построим график заданной функции. Минимум в точке $x=0.315$



Результат работы программы

```
Variant 10:
Dichotomy search:
-2 4 6 9.13534 63.5982
-2 0.99 2.99 9.13534 2.69133
-0.495 0.99 1.485 2.8446 2.69133
0.2575 0.99 0.7325 1.845 2.69133
0.2575 0.61375 0.35625 1.845 1.99654 |
0.2575 0.425625 0.168125 1.845 1.86045
0.2575 0.331563 0.0740625
l<epsilon
x = 0.294531 +- 0.0370312
Passive search:
N: 1 Minimum: -2 +- 3
N: 2 Minimum: 0 +- 2
N: 3 Minimum: 0 +- 1.5
N: 4 Minimum: 0.4 +- 1.2
N: 5 Minimum: 0.4 +- 1
N: 6 Minimum: 0.4 +- 0.857143
N: 7 Minimum: 0.25 +- 0.75
N: 8 Minimum: 0.25 +- 0.666667
N: 9 Minimum: 0.25 +- 0.6
N: 10 Minimum: 0.25 +- 0.545455
N: 11 Minimum: 0.25 +- 0.5
N: 12 Minimum: 0.307692 +- 0.461538
N: 13 Minimum: 0.307692 +- 0.428571
N: 14 Minimum: 0.307692 +- 0.4
N: 15 Minimum: 0.307692 +- 0.375
N: 16 Minimum: 0.307692 +- 0.352941
N: 17 Minimum: 0.307692 +- 0.333333

N: 38 Minimum: 0.307692 +- 0.133040
N: 39 Minimum: 0.307692 +- 0.15
N: 40 Minimum: 0.307692 +- 0.146341
N: 41 Minimum: 0.307692 +- 0.142857
N: 42 Minimum: 0.307692 +- 0.139535
N: 43 Minimum: 0.318182 +- 0.136364
N: 44 Minimum: 0.318182 +- 0.133333
N: 45 Minimum: 0.318182 +- 0.130435
N: 46 Minimum: 0.318182 +- 0.12766
N: 47 Minimum: 0.318182 +- 0.125
N: 48 Minimum: 0.318182 +- 0.122449
N: 49 Minimum: 0.318182 +- 0.12
N: 50 Minimum: 0.318182 +- 0.117647
N: 51 Minimum: 0.318182 +- 0.115385
N: 52 Minimum: 0.318182 +- 0.113208
N: 53 Minimum: 0.318182 +- 0.111111
N: 54 Minimum: 0.318182 +- 0.109091
N: 55 Minimum: 0.318182 +- 0.107143
N: 56 Minimum: 0.315789 +- 0.105263
N: 57 Minimum: 0.315789 +- 0.103448
N: 58 Minimum: 0.315789 +- 0.101695
N: 59 Minimum: 0.315789 +- 0.1
```

4. Выводы

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

Ссылка на git-репозиторий: https://github.com/AnnaMinkova/Tsisa_lab_01

Приложение 1. Исходный код программы

```
#include<iostream>

#include <cmath>
using namespace std;
double myFunction(double x)
{
    return pow((1-x),2)+exp(x);
}
void Print1(double ak, double bk)
{
    cout<<ak<< ' ' <<bk<< ' ' <<bk-ak<< ' ' <<myFunction(ak)<< ' '
<<myFunction(bk)<< ' ' <<endl;
}

const double A=-2.;
const double B=4.;
const double Epsilon=0.1;

void Dichotomy(double a, double b)
{
    cout<<"Dichotomy search: "<<endl;
    const double Delta=.01;
    while (b-a>Epsilon)
    {
        Print1(a,b);
        double x1 = a+(b-a)/2-Delta, x2 = a+(b-a)/2+Delta;
        myFunction(x1) < myFunction(x2)
        ? b=x1
        : a=x2;
    }
    cout<<a<<' ' <<b<<' ' <<b-a<<' ' <<"\n\epsilon" <<"\nx = "<<(a+b)/2<<" +-
"<<(a+b)/2-a<<endl;
}
int main()
{
    cout<<"Variant 10:"<<endl;
    Dichotomy(A,B);
    cout<<"Passive search: "<<endl;
    int N=1,a=-2,b=4;
    double eps,x,y;
    x=a;
    y=myFunction(x);
    eps=(b-a)/(N+1);

    while (eps>=0.1)
```

```

{

    for (int k=1;k<N;k++)
    {
        double xk;
        xk=(double)(b-a)*k/(N+1)+a;
        if (myFunction(xk)<=y)
        {
            y=myFunction(xk);
            x=xk;
        }

    }
    cout<<"N: "<<N<<" ";
    cout<<"Minimum: "<<x<<" +- "<<eps<<endl;
    N++;
    eps=(double)(b-a)/(N+1);
}

}

```