|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 16.

Выполнил: Сердюкова М.Ю.,

студент группы ИУ8-32

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) на примере унимодальной функции одного переменного.

# 2. Постановка задачи

Унимодальная функция:

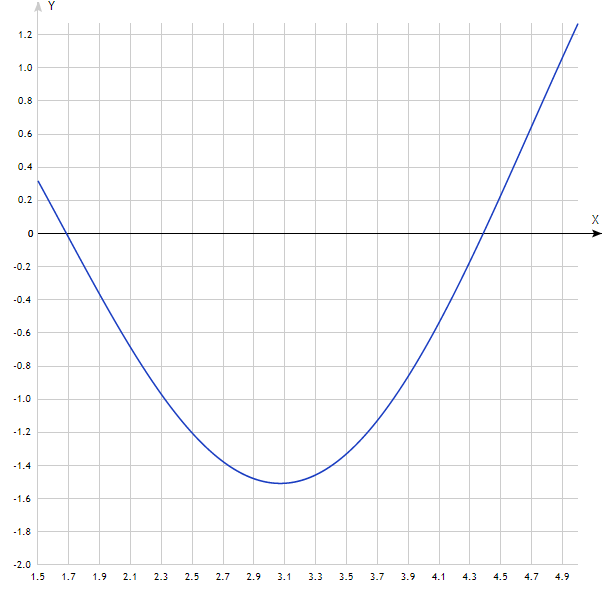
Отрезок поиска:

Методы поиска: оптимальный пассивный, дихотомия.

Наибольшая длина интервала неопределенности:

# 3. Ход работы

Рисунок 1 демонстрирует график унимодальной функции.



**Рисунок 1** – график функции.

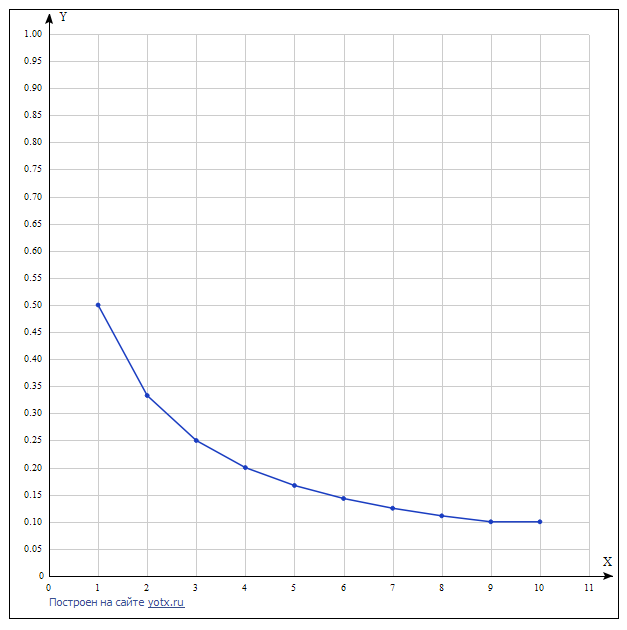
**Реализация оптимального пассивного поиска.**

Точки расположены равномерно по отрезку, следовательно, координата точки с номером :

**Таблица 1** – применение оптимального пассивного поиска

|  |  |
| --- | --- |
| Количество точек (N) | Значение x в минимуме |
| 1 | 11.50 ± 0.5 |
| 2 | 10.67 ± 0.333 |
| 3 | 10.25 ± 0.25 |
| 4 | 11.00 ± 0.2 |
| 5 | 10.67 ± 0.167 |
| 6 | 11.14 ± 0.143 |
| 7 | 10.88 ± 0.125 |
| 8 | 10.67 ± 0.111 |
| 9 | 11.00 ± 0.1 |

Минимальное значение функции достигается при N = 10 (x = 11 ± 0.1)



**Рисунок 2** – график зависимости погрешности от числа измерений (первые 10) метода опт. пассивного поиска.

**Реализация метода дихотомии.**

Отрезок поиска делится пополам точкой x. Вычисляются значения функции на границах окрестности точки x: . Исключается левая половина, если значение функции в точке левой границы окрестности больше, чем в правой. Иначе, исключается правая половина. Эти действия повторяются до тех пор, пока отрезок поиска будет больше отрезка неопределенности.

**Таблица 2** – последовательный поиск (метод дихотомии)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Начало интервала (ak) | Конец интервала (bk) | Длина интервала 1 | f(ak) | f(bk) |
| 9 | 14 | 5 | 122.311 | 527.802 |
| 9 | 11.5 | 2.5 | 122.311 | 55.1051 |
| 10.25 | 11.5 | 1.25 | 50.3932 | 55.1051 |
| 10.25 | 10.875 | 0.625 | 50.3932 | 43.8245 |
| 100.5625 | 10.875 | 0.3125 | 45.0125 | 43.8245 |
| 10.7188 | 10.875 | 0.15625 | 43.9064 | 43.8245 |
| 10.7969 | 10.875 | 0.078125 | 43.7368 | 43.8245 |

Минимальное значение функции достигается при x = 10.8359±0.04,

# 4. Вывод

В данной лабораторной работе был найден минимум унимодальной функции с помощью метода оптимального пассивного поиска и метода дихотомии. Из приведенного выше хода работы можно сделать вывод, что метод дихотомии эффективнее метода пассивного поиска, т.к. требует меньшего количества контрольных точек.

# Приложение 1. Код программы main.cpp.

#include <iostream>

#include <math.h>

using namespace std;

void table\_ditoch(const double& ak,const double& bk){

std::cout << '|';

std::cout.width(14);\

std::cout.setf(std::ios::left);

std::cout.fill(' ');

std::cout << ak;

std::cout << "| ";

std::cout.width(10);

std::cout.fill(' ');

std::cout<<bk;

std::cout << "| ";

std::cout.width(10);

std::cout.fill(' ');

std::cout<<bk-ak;

std::cout << "| ";

std::cout.width(10);

std::cout.fill(' ');

std::cout<<pow(ak,2)\*exp(sin(ak));

std::cout << "| ";

std::cout.width(10);

std::cout.fill(' ');

std::cout<<pow(bk,2)\*exp(sin(bk));

std::cout<<"|\n";

}

void table\_passive (const double mini, const double i){

std::cout.setf(std::ios::left);

std::cout.precision(0);

std::cout<<"| "<<i<<" | ";

std::cout.precision(2);

std::cout.width(2);

std::cout<<std::fixed<<mini<<" ± ";

std::cout.width(7);

std::cout.fill(' ');

std::cout.precision(3);

std::cout.unsetf(std::ios\_base::fixed);

std::cout << 1./(i+1);

std::cout<<"|\n";

}

double f(const double &x){

double f = pow(x, 2) \* exp(sin(x));

return f;

}

int main() {

setlocale(LC\_ALL, "Russian");

int a=9, b=14;

const double e=0.1;

double x1, x2, ak=a, bk=b, s=0.04;

std::cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

std::cout<<"| Начало | Конец | Длина | | |\n";

std::cout<<"| интервала | интервала | интервала | f(ak) | f(bk) |\n";

std::cout<<"| (ak) | (bk) | 1 | | |\n";

std::cout<<"------------------------------------------------------------------------\n";

while (bk-ak>e){

table\_ditoch(ak,bk);

x1=(ak+bk)/2.-s;

x2=(ak+bk)/2.+s;

if (f(x1)<=f(x2)) bk=x2-s;

else ak=x1+s;

}

table\_ditoch(ak, bk);

std::cout<<"------------------------------------------------------------------------\n";

std::cout<<'\n';

std::cout<<"Минимальное значение функции достигается при x = "<<(ak+bk)/2.<<"±"<<s<<'\n';

std::cout<<'\n';

size\_t j;

double i=1;

double min, part, mini;

size\_t N=(1/e)-1;

std::cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

std::cout<<"| Количество | Значение x |\n";

std::cout<<"| точек (N) | в минимуме |\n";

std::cout<<"-----------------------------\n";

for (i; i<=N; i++){

part=(b-a)/(i+1);

min=10000;

for (j=1; j<=i; j++) {

if (f(a + part \* j) < min) {

min = f(a + part \* j);

mini = a + part \* j;

}

}

table\_passive(mini, i);

}

std::cout<<"-------------------------------\n\n";

std::cout<<"Минимальное значение функции достигается при N = "<<i<<" (x = "<<mini<<" ± ";

std::cout.precision(1);

std::cout<<std::fixed<<1./(i+1)<<")\n";

return 0;

}

**Ссылка на git-репозиторий:** https://github.com/SerdukovaM/TS\_lab1