|  |  |
| --- | --- |
| 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:

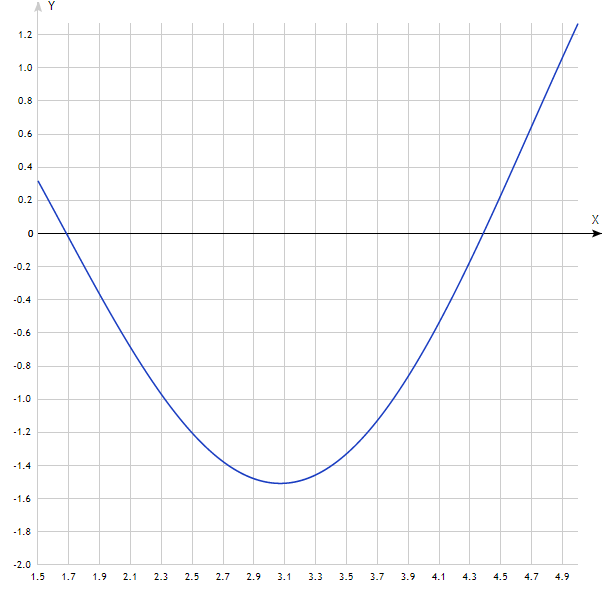
Отрезок поиска:

Методы поиска: оптимальный пассивный, дихотомия.

Наибольшая длина интервала неопределенности:

# 3. Ход работы

Рисунок 1 демонстрирует график унимодальной функции.



**Рисунок 1** – график функции.

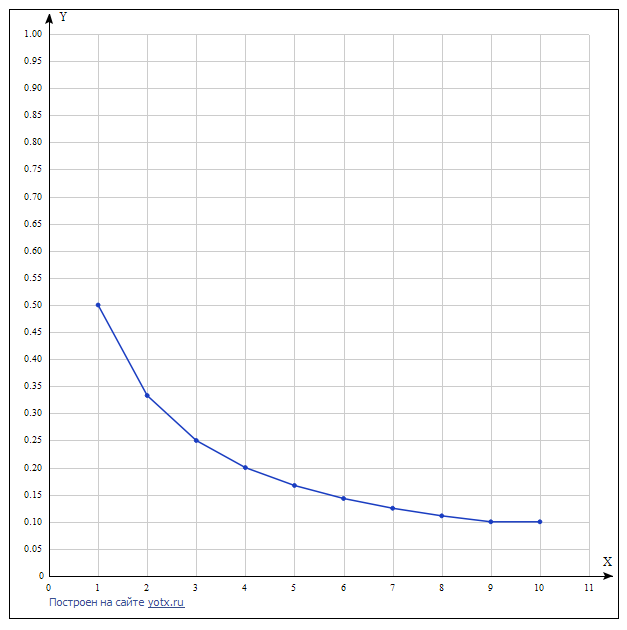
**Реализация оптимального пассивного поиска.**

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;

double f(const double &x) {

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

return f;

}

double fm(const double &x) {

double f = pow(x, 2)\* exp(sin(x))\* (sin(5 \* x));

return f;

}

void table(std::string type, int N[19][10]) {

int i = 0, j = 0;

std::cout << "+-------+------------+------------+------------+------------+-------"

"-----+------------+------------+-----------"

"-+------------+------------+\n";

std::cout << "| q/P | 0.9 | 0.91 | 0.92 | 0.93 | "

"0.94 | 0.95 | 0.96 | "

" 0.97 | 0.98 | 0.91 |\n";

std::cout << "+-------+------------+------------+------------+------------+-------"

"-----+------------+------------+-----------"

"-+------------+------------+\n";

double min = 9;

for (i = 0; i < 19; i++) {

std::cout << "| ";

std::cout.width(6);

std::cout << (i + 1) \* 0.005 << "| ";

for (j = 0; j < 14; j++) {

min = 9;

for (int t = 0; t < N[i][j]; t++) {

double x = ((double)rand() / RAND\_MAX) \* 3 + 7;

if (type == "унимодальная") {

if (f(x) < f(min))

min = x;

} else {

if (fm(x) < fm(min))

min = x;

}

}

std::cout.width(11);

if (type == "унимодальная") {

std::cout << f(min) << "| ";

} else {

std::cout << fm(min) << "| ";

}

}

std::cout << '\n';

}

std::cout << "+-------+------------+------------+------------+------------+-------"

"-----+------------+------------+-----------"

"-+------------+------------+\n\n";

}

int main() {

setlocale(LC\_ALL, "Russian");

int N[19][10];

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

std::cout << "+-------+------+------+------+------+------+------+------+-----"

"-+------+------+\n";

std::cout << "| q/P | 0.9 | 0.91 | 0.92 | 0.93 | 0.94 | 0.95 | 0.96 | "

"0.97 | 0.98 | 0.99 |\n";

std::cout << "+-------+------+------+------+------+------+------+------+-----"

"-+------+------+\n";

int i = 0, j = 0;

for (double q = 0.005; q <= 0.1; q += 0.005) {

std::cout << "| ";

std::cout.width(6);

std::cout << q << "| ";

j = 0;

for (double P = 0.9; P < 1; P += 0.01) {

N[i][j] = int((log(1 - P)) / (log(1 - q)));

std::cout.width(5);

std::cout << N[i][j] << "| ";

j++;

}

std::cout << '\n';

i++;

}

std::cout << "+-------+------+------+------+------+------+------+------+-----"

"-+------+------+\n\n";

table("унимодальная", N);

table("модальная", N);

}

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