**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 № 2

По дисциплине: «Теория систем и системный анализ»

# Тема: «Исследование метода случайного поиска экстремума

# функции одного переменного»

# Вариант 6

Выполнил: Гуща Н.В.,

студент группы ИУ8-32

Проверил: Коннова Н.С.

Доцент каф. ИУ8

г. Москва 2020 г.

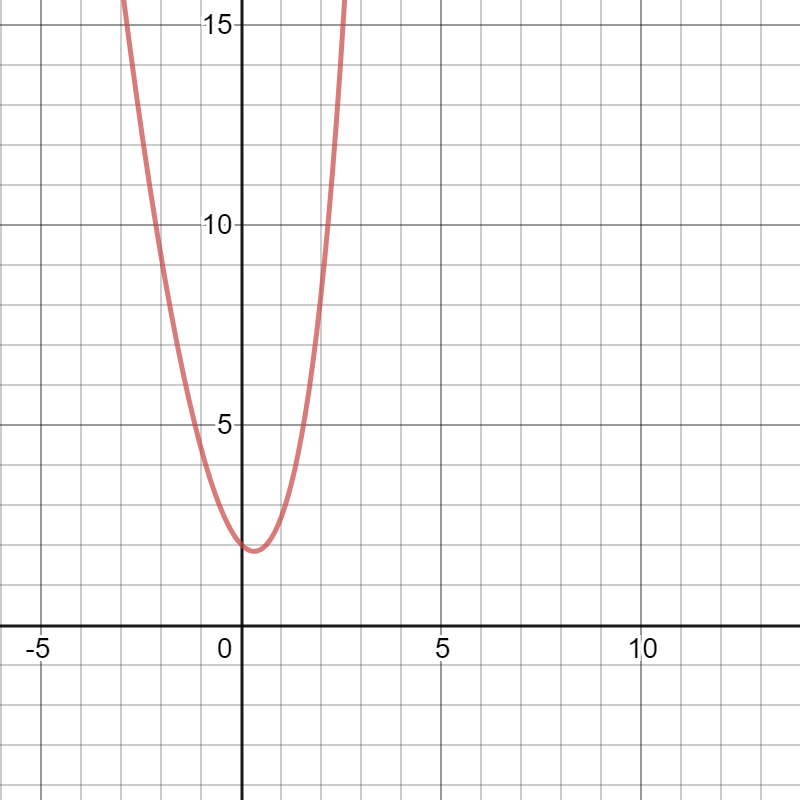
**Цель:**

Изучение метода случайного поиска экстремума на примере унимодальной и мультимодальной функций одного переменного.

**Условие задачи:**

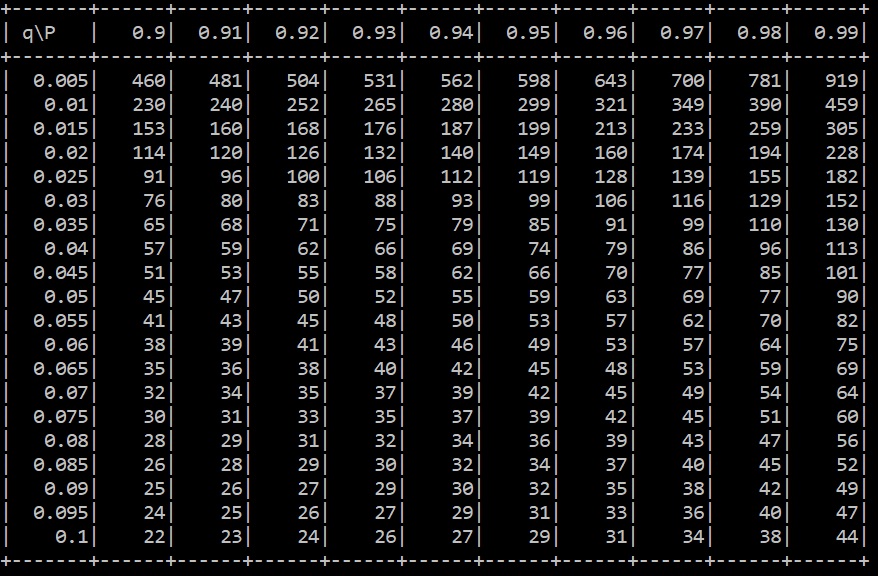
1.На интервале [-5;2] задана унимодальная функция одного переменного f(x)=(1-x)^2+exp(x). Используя метод случайного поиска осуществить поиск минимума f(x) с заданной вероятностью попадания в окрестность экстремума P при допустимой длине интервала неопределенности ε. Определить необходимое число испытаний N. Численный эксперимент выполнить для значений P=0,90,0,91,…,0,99 и значений ε=(b-a)q, где q=0,005,0,010,…,0,100.

2. При аналогичных исходных условиях осуществить поиск минимума f(x), модулированной сигналом sin5x, т.е. мультимодальной функции f(x)\*sin5x.

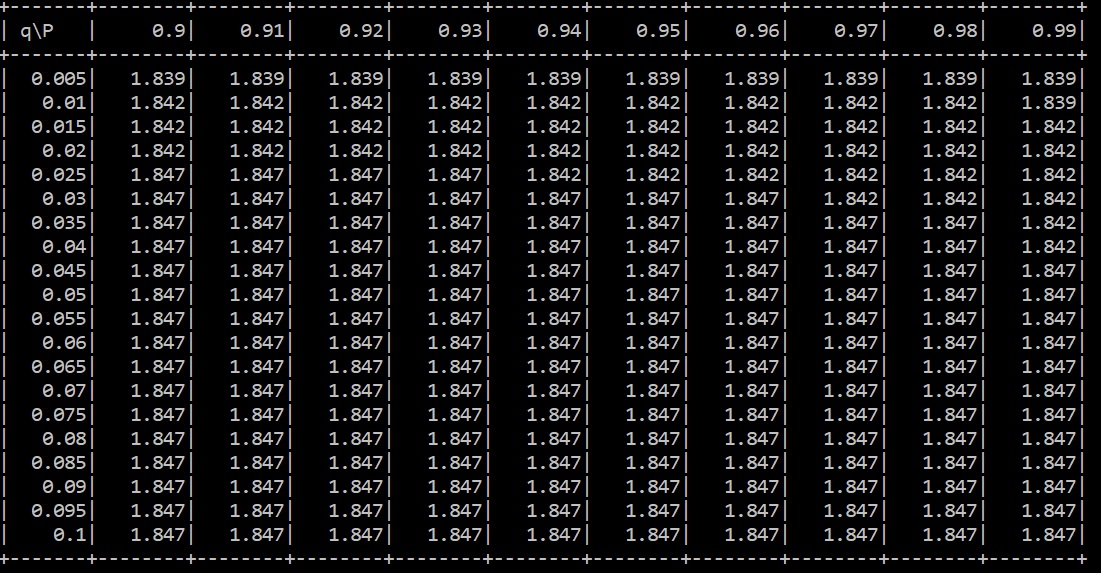
****

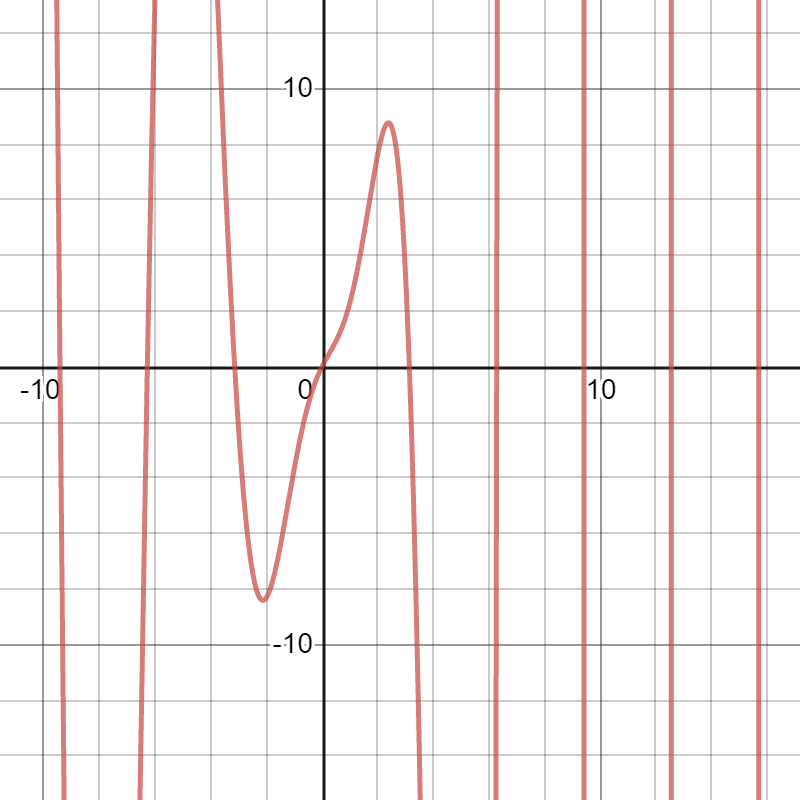
**Рисунок 1.** График функции f(x)=(1-x)^2+exp(x) на интервале [-5;2].

**Таблица 1.** Зависимость N от P и q.



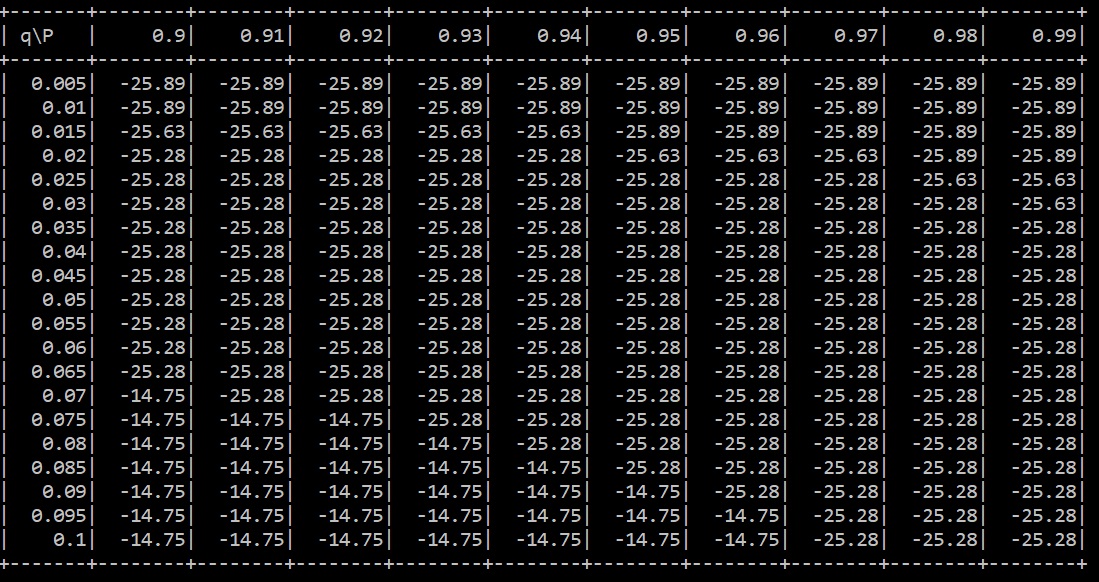
**Таблица 2.** Результаты поиска экстремума f(x) в зависимости от P и q.





**Рисунок 2.** График функции f(x)\*sin5x на интервале [-5;2].

**Таблица 3.** Результаты поиска экстремума f(x)\*sin5x в зависимости от P и q.



**Код программы:**

//Кафедра "Информационная безопасность"

//ИУ8-32

//Гуща Н.В.

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

#include <iostream>

#include <math.h>

#include <iomanip>

#include <vector>

#include <iostream>

#include <iomanip>

#include <vector>

#include <math.h>

#include <algorithm>

using namespace std;

const double a = -5.0, b = 2.0;

double func(double x) {

return (1-x)\*(1-x)+exp(x);

}

void separator(int lhs, int rhs) {

cout << "+";

cout << setfill('-') << setw(lhs) << "+";

for (int i = 0; i < 10; i++) {

cout << setw(rhs) << "+";

}

cout << endl;

}

void printfirsttable(double q, double P) {

separator(8, 7);

cout << "| q\\P |";

cout << setfill(' ');

for (int i = 0; i < 10; i++) {

cout << setw(6) << P + i \* 0.01 << "|";

}

cout << endl;

separator(8, 7);

cout << setfill(' ');

for (q = 0.005; q < 0.105;) {

cout << "|" << setw(7) << q << "|";

for (P = 0.9; P < 1;) {

double N = log(1 - P) / log(1 - q);

cout << setw(6) << ceil(N) << "|";

P += 0.01;

}

cout << endl;

q += 0.005;

}

separator(8, 7);

}

long double min(double q, double P) {

srand(time(NULL));

int N = ceil(log(1 - P) / log(1 - q));

vector<long double> val;

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

double x = a + (b - a) \* rand() / RAND\_MAX;

val.push\_back(func(x));

}

return \*min\_element(val.begin(), val.end());

}

void get\_min(double q, double P) {

separator(8, 9);

cout << "| q\\P |";

cout << setfill(' ');

for (int i = 0; i < 10; i++) {

cout << setw(8) << P + i \* 0.01 << "|";

}

cout << endl;

separator(8, 9);

cout << setfill(' ');

for (q = 0.005; q < 0.105;) {

cout << "|" << setw(7) << q << "|";

for (P = 0.9; P < 1;) {

cout << setw(8) << setprecision(4) << min(q, P) << "|";

P += 0.01;

}

cout << endl;

q += 0.005;

}

separator(8, 9);

}

long double min\_sin(double q, double P) {

srand(time(NULL));

int N = ceil(log(1 - P) / log(1 - q));

vector<long double> val;

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

double x = a + (b - a) \* rand() / RAND\_MAX;

val.push\_back(func(x) \* sin(5 \* x));

}

return \*min\_element(val.begin(), val.end());

}

void get\_min\_sin(double q, double P) {

separator(8, 9);

cout << "| q\\P |";

cout << setfill(' ');

for (int i = 0; i < 10; i++) {

cout << setw(8) << P + i \* 0.01 << "|";

}

cout << endl;

separator(8, 9);

cout << setfill(' ');

for (q = 0.005; q < 0.105;) {

cout << "|" << setw(7) << q << "|";

for (P = 0.9; P < 1;) {

cout << setw(8) << setprecision(4) << min\_sin(q, P) << "|";

P += 0.01;

}

cout << endl;

q += 0.005;

}

separator(8, 9);

}

int main() {

double q = 0.005, P = 0.9;

printfirsttable(q, P);

cout << endl;

get\_min(q, P);

cout << endl;

get\_min\_sin(q, P);

return 0;

}

**Вывод**

Из полученных таблиц и графиков видно, что применимость метода случайного поиска не зависит от того, является ли функция унимодальной и мультимодальной. Для уменьшения интервала неопределенности необходимо увеличивать число случайных точек N.

**Контрольные вопросы**

1.В чем состоит сущность метода случайного поиска? Какова область применимости данного метода?

Сущность метода состоит в выборе случайных точек по равномерному закону, дальнейшем запоминании значений функции во всех выбранных точках и присвоении решению наилучшей из выбранных точек(минимум или максимум из найденных значений функции в зависимости от условия задачи).

Метод случайного поиска применим как к унимодальным, так и к мультимодальным функциям.

2.Поясните принцип разбиения интервала при случайном поиске.

Имеем n-мерное пространство, объём n-мерного прямоугольника, в котором ведётся поиск минимума

Если необходимо найти решение с точностью , по каждой из переменных, то мы должны попасть в окрестность оптимальной точки с объемом . Тогда данное пространство разбивается на меньшие подпространства, ограниченные уже заданной точностью . Далее просчитывается вероятность попадания в эту окрестность в зависимости от количества экспериментов N.

3.Что такое интервал неопределенности? Приведите выражения для оценки интервала неопределенности для метода случайного поиска.

Интервал неопределенности – интервал, в пределах которого находится искомое значение функции(для n-мерного пространства n-мерный параллелепипед).

– объем n-мерного параллелепипеда, внутри которого находится искомое значение функции.