|  |  |
| --- | --- |
| 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 № 2

по дисциплине «Теория Систем и Системный Анализ»

**Тема: «Исследование метода случайного поиска экстремума**

**функции одного переменного»**

Вариант 4

Выполнил: Велинецкий А. В.,

студент группы ИУ8-32

Проверил: Коннова Н. С.,

доцент каф. ИУ8

г. Москва,

2019 г.

# 1. Цель работы

Изучить методы случайного поиска экстремума на примере унимодальной и мультимодальной функций одного переменного.

# 2. Постановка задачи

Унимодальная функция: f(x) =

Отрезок поиска:

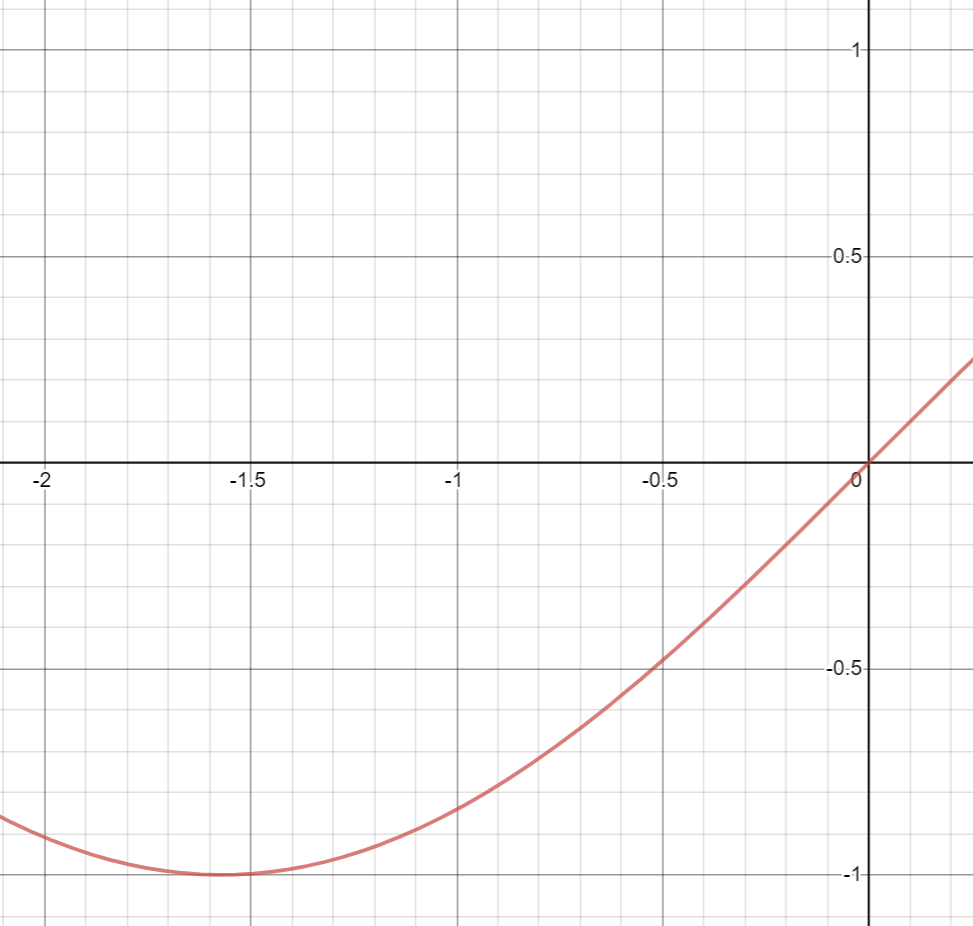
Используя метод случайного поиска осуществить поиск минимума f (x) с заданной вероятностью попадания в окрестность экстремума P при допустимой длине интервала неопределенности ε. Определить необходимое число испытаний N. Численный эксперимент выполнить для значений P = 0,90, 0,91, ... , 0,99 и значений ε = (b-a)q , где q = 0,005, 0,010,..., 0,100

При аналогичных исходных условиях осуществить поиск минимума f (x),

модулированной сигналом sin(5x), т.е. мультимодальной функции f(x) \*sin(5x).

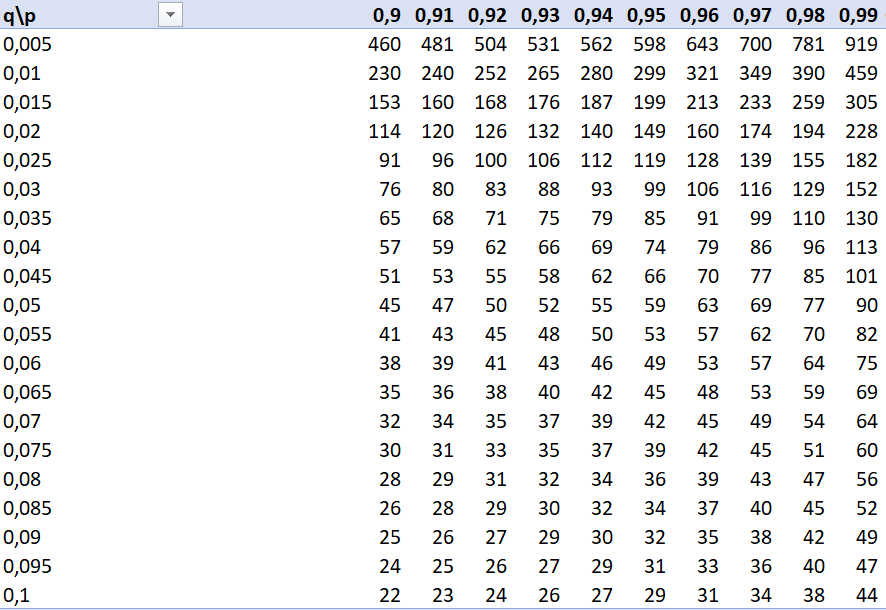
# 3. Ход работы

Рисунок 1 демонстрирует график унимодальной функции.



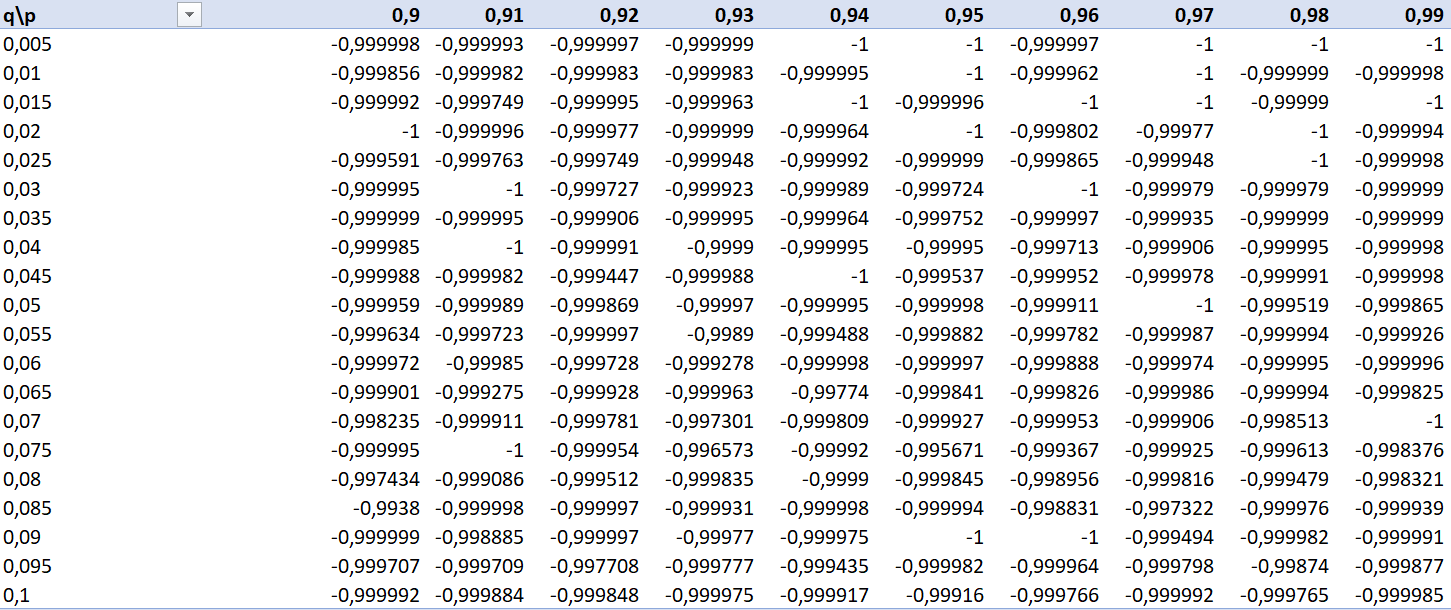
**Рисунок 1** – график функции f(x)

Посчитаем количество измерений N (Вероятность непопадания в интервал неопределенности за N испытаний) при заданных значениях P и q (В данном методе P – это вероятность того, что найденная точка минимума находится в интервале неопределенности, а q – это вероятность попадания в интервал неопределенности для отдельно взятой точки).



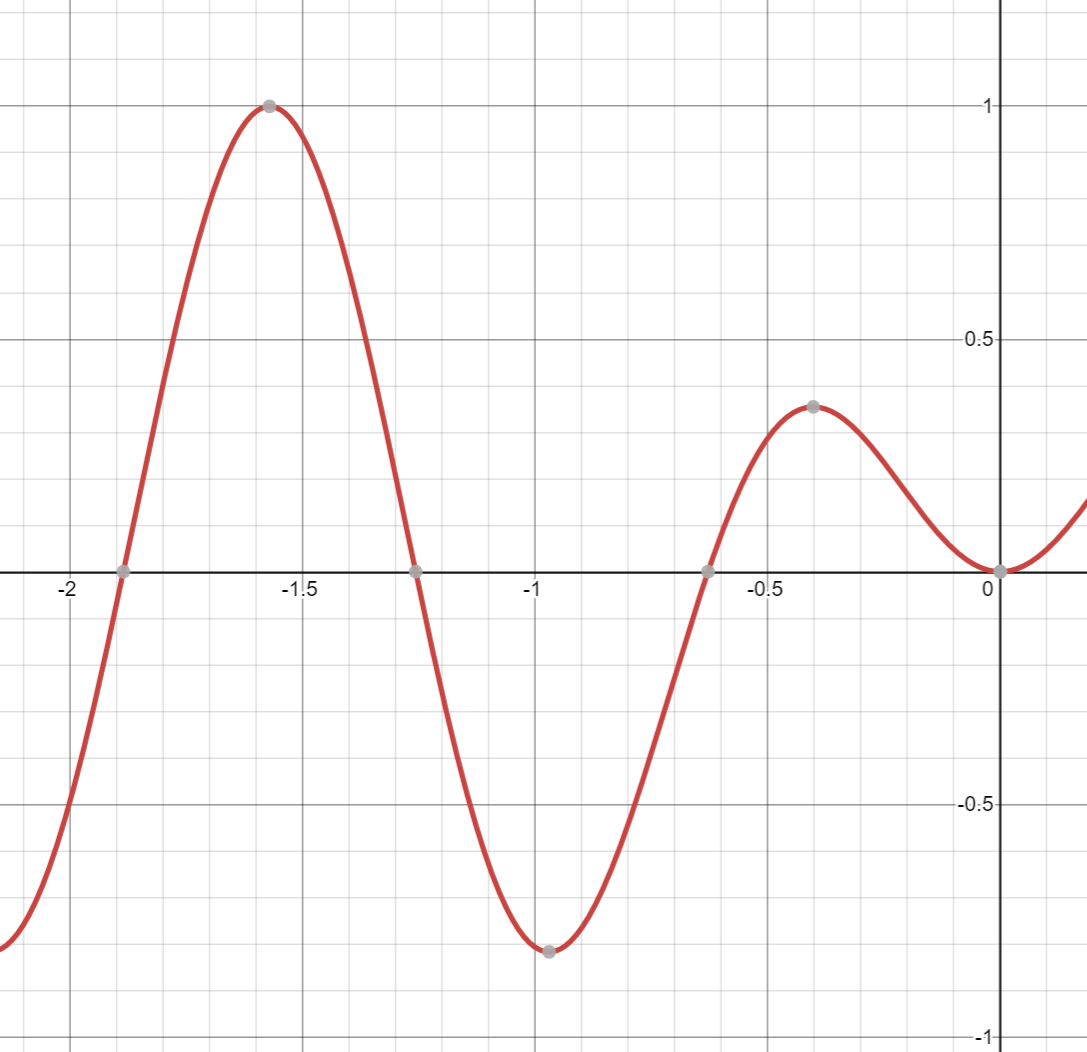
**Таблица 1** – Количество измерений N

Далее найдем экстремум нашей функции.

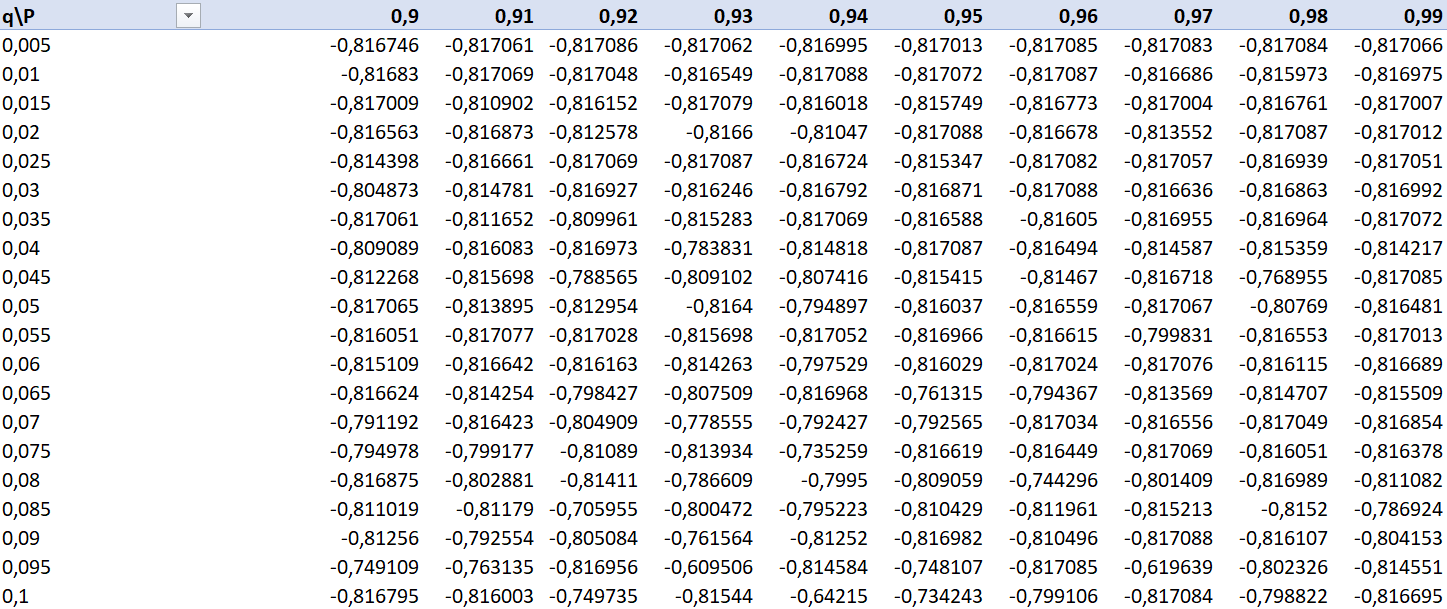


**Таблица 2** – Экстремумы f(x)

После этого найдем экстремум функции f(x)\*sin(5x):



**Рисунок 2** – график функции f(x)\*sin(5x)



**Таблица 2** – Экстремумы f(x)\*sin(5x)

# 4. Выводы

В ходе данной лабораторной работы мы изучили методы случайного поиска экстремума на примере унимодальной и мультимодальной функций одного переменного. В результате удалось выяснить, что при использовании данного метода не имеет значение, функция является унимодальной или мультимодальной на промежутке. Для увеличения вероятности попадания в заданный интервал или для уменьшения интервала неопределенности необходимо увеличивать число случайных точек.

# 5. Контрольные вопросы

# Приложение 1. Исходный код программы. Файл main.cpp

#include <cmath>  
#include <vector>  
#include <iostream>  
#include <algorithm>  
  
using namespace std;  
  
double myfun(double x) {  
 return (cos(x)\*tan(x));  
}  
  
double myfun\_sin5x(double x) {  
 return sin(5 \* x) \* myfun(x);  
}  
  
int N\_count(double q, double P) {  
 return (ceil(log(1. - P) / log(1. - q)));  
}  
  
double Random(int a, int b)  
{  
 return a + (1.\*rand()/RAND\_MAX)\*(b - a); // (1.\*rand()/RAND\_MAX)-генерация числа от 0 до 1  
}  
  
int main() {  
 int a = -2;  
 int b = 0;  
 int N;  
 double MAX\_DOUBLE = 1.7976931348623158\*pow(10, 308);;  
 double x;  
 double q = 0.005;  
 for (int i = 0; i < 20; i++) {  
 double p = 0.9;  
 for (int j = 0; j < 10; j++) {  
 N = N\_count(q, p);  
 double min = MAX\_DOUBLE;  
 for (auto k = 0; k < N; k++) {  
 x = Random(a, b);  
 if (myfun(x) < min) min = myfun(x);  
 }  
 cout << "P = " << p << " q = " << q  
 << " N = " << N << " min: " << min << endl;  
 p+=0.01;  
 }  
 q+=0.005;  
 }  
 cout << endl;  
 q = 0.005;  
 for (auto i = 0; i < 20; i++) {  
 double p = 0.9;  
 for (auto j = 0; j < 10; j++) {  
 N = N\_count(q, p);  
 double min = MAX\_DOUBLE;  
 for (auto k = 0; k < N; k++) {  
 x = Random(a, b);  
 if (myfun\_sin5x(x) < min) min = myfun\_sin5x(x);  
 }  
 cout << "P = " << p << " q = " << q  
 << " N = " << N << " min: " << min << endl;  
 p+=0.01;  
 }  
 q+=0.005;  
 }  
 return 0;  
}