|  |  |
| --- | --- |
| 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 16.

Выполнил: Сердюкова М.Ю.,

студентка группы ИУ8-33

Проверил: Коннова Н. С.,

доцент каф. ИУ8

г. Москва,

2020 г.

# 1. Цель работы

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

# 2. Постановка задачи

Унимодальная функция:

Отрезок поиска:

1. На интервале [a,b] задана унимодальная функция одного переменного f(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.

Последовательность действий:

- определить вероятность P1 непопадания в ε -окрестность экстремума за одной испытание;

- записать выражение для вероятности Pn непопадания в ε -окрестность экстремума за N испытаний;

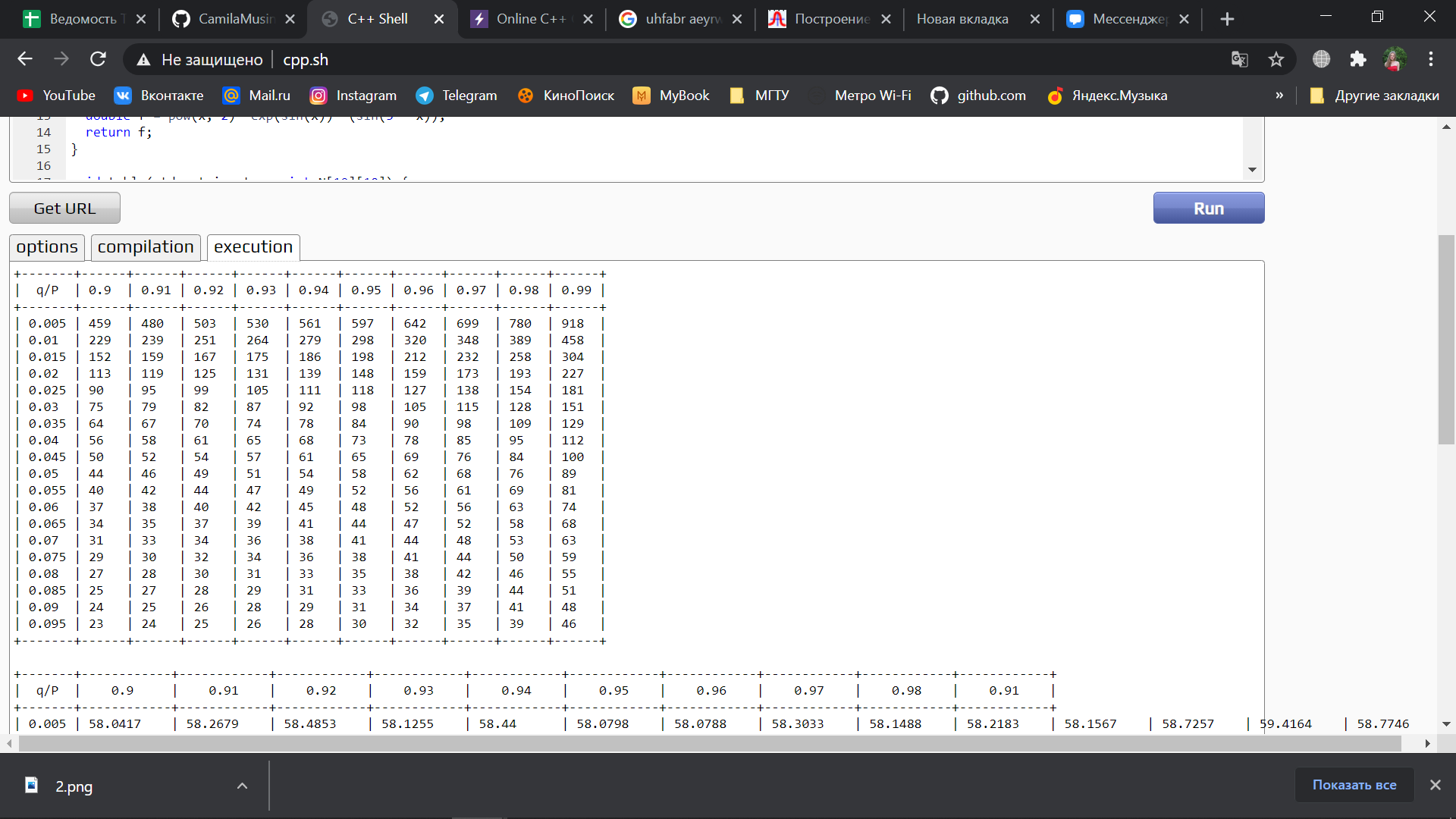
- из выражения для Pn определить необходимое число испытаний N в зависимости от заданных Pn = P и ε.

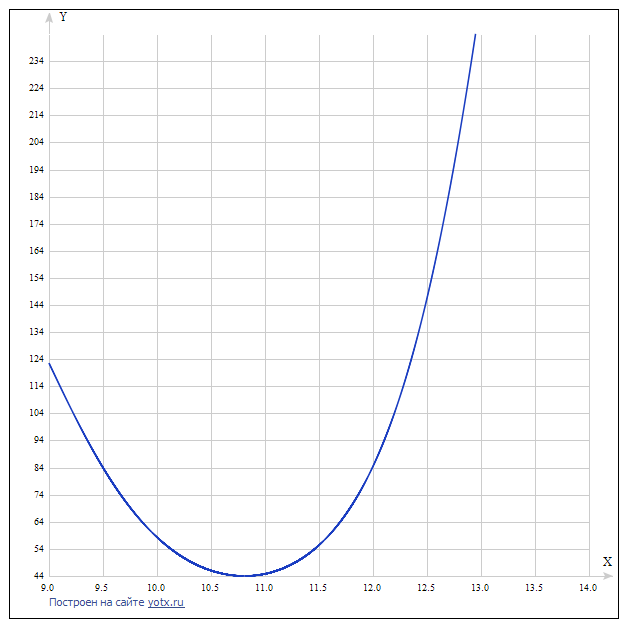
2. При аналогичных исходных условиях осуществить поиск минимума f(x), модулированной сигналом sin5x , т.е. мультимодальной функции f(x)sin5.

# 3. Ход работы

Задана функция: f(x) = на интервале [9,14] (см. рисунок 1). 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. Тогда вероятность непопадания в интервал неопределенности за одно испытание будет равна 1-q. Вероятность непопадания в интервал неопределенности за N испытаний будет равна (1-q)^N. Тогда вероятность P = 1-(1-q)^N. Отсюда можем найти N = ln(1-P)/ln(1-q). Случайно выбираем N точек в заданном отрезке [a,b], определим значение унимодальной функции в этих точках и среди них найдем наименьшее значение. Результаты численного эксперимента для представим в виде таблицы в зависимости от P и q.

**Таблица 1** – зависимость N от P и q

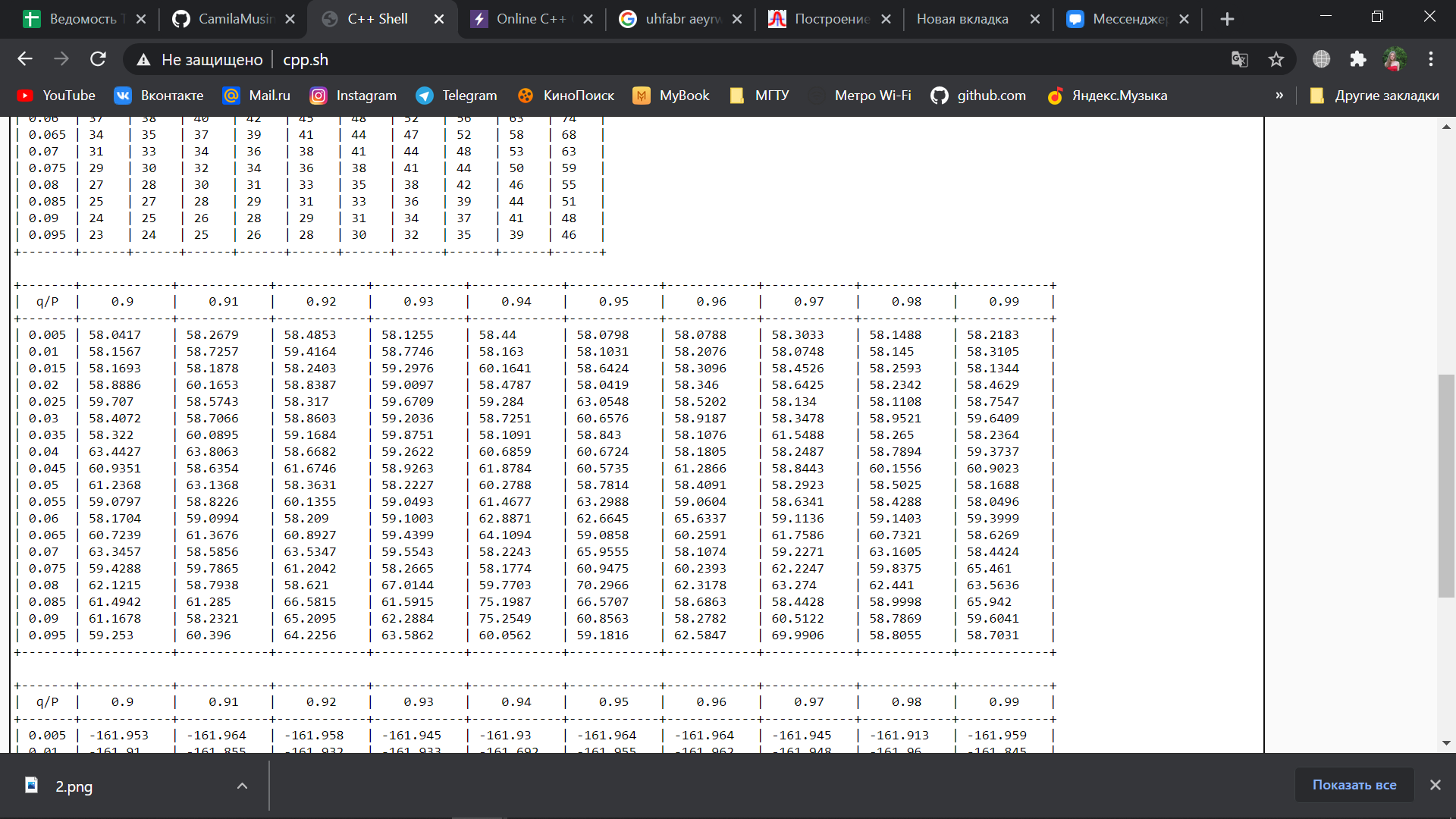


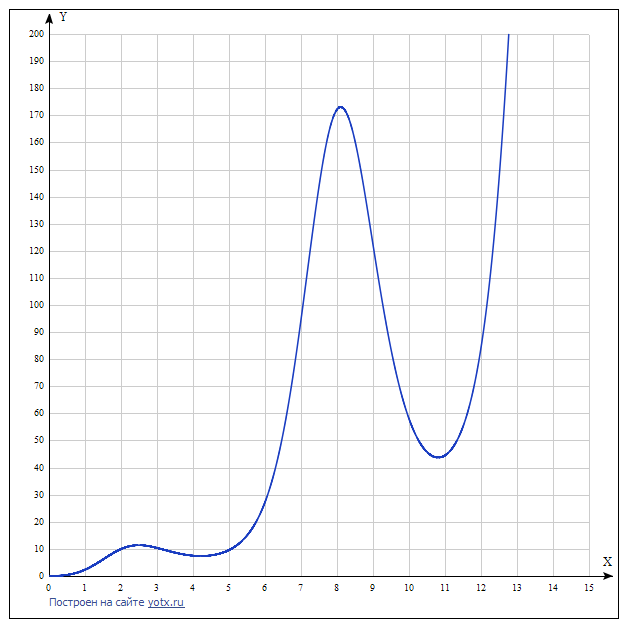


**Рисунок 1** – график функции f(x)

Рисунок 1 демонстрирует график унимодальной функции.

**Таблица 2** – результаты поиска экстремума f(x) в зависимости от P и q

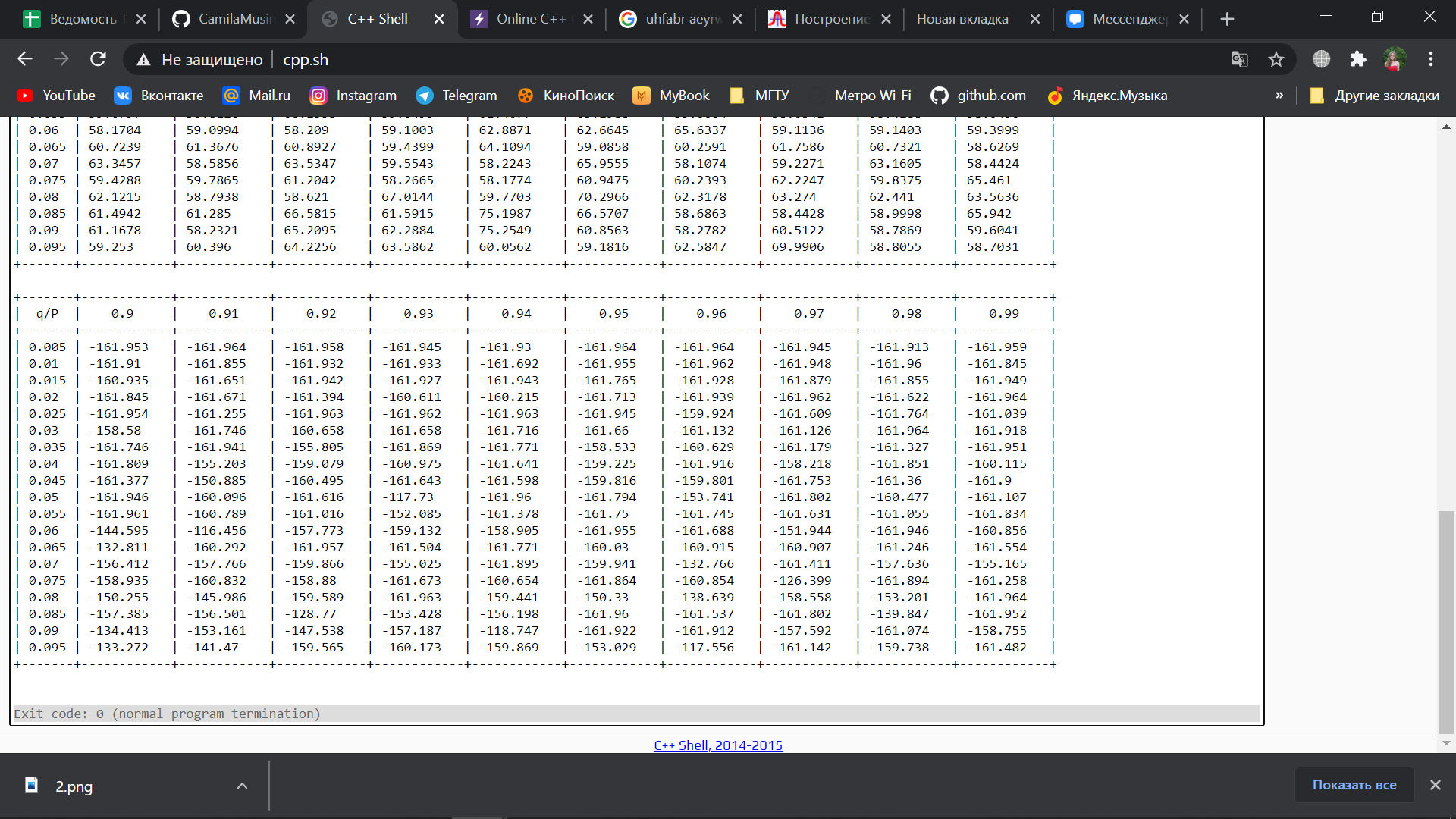




**Рисунок 2** – график функции f(x)\*sin(5x)

Рисунок 2 демонстрирует график мультимодальной функции.

**Таблица 2** – результаты поиска экстремума f(x)\*sin(5x) в зависимости от P и q



# 4. Выводы

Как видно из полученных результатов, применимость метода случайного поиска не зависит от того, является ли функция унимодальной или мультимодальной. Для увеличения вероятности попадания в заданный интервал или для уменьшения интервала неопределенности необходимо увеличивать число случайных 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 | 0.99 |\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 < 10; 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\_lab2