****

**Министерство науки и высшего образования Российской Федерации**

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. Н. Э. БАУМАНА**

Факультет: «Информатика и системы управления»

Кафедра: «Информационная безопасность (ИУ8)» / ГУИМЦ

**ТЕОРИЯ СИСТЕМ И СИСТЕМНЫЙ АНАЛИЗ**

**Лабораторная работа №2 на тему:**

«Исследование методы случайного поиска экстремума функции одного переменного»

Вариант 4

**Преподаватель:** Коннова Н. С.

**Студент:** Месяцева Н. В.

**Группа**: ИУ8Ц-51

Москва

2019

# **Цель работы**

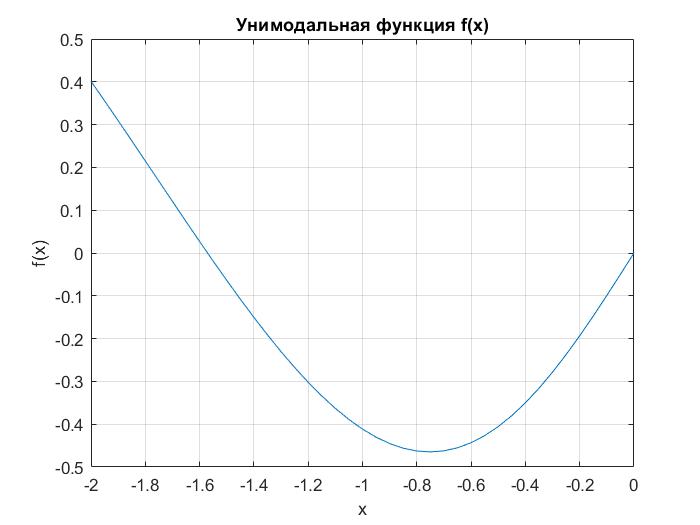
Изучение методы случайного поиска экстремума на примере унимодальной и мультимодальной функции одного переменного.

# **Постановка задачи**

1. На интервале задана унимодальная функция одного переменного . Используя метод *случайного поиска,* осуществить поиск минимума с заданной вероятностью попадания в окрестность экстремума при допустимой длине интервала неопределенности . Определить необходимое число испытаний . Численный эксперимент выполнить для значений , где

2. При аналогичных исходных условиях осуществить поиск минимума , модулированной сигналом , т.е. *мультимодальной* функции

# **Ход работы**

Известно, что на интервале дана унимодальная функция и мультимодальная функция Ниже представлены графики данных функций.

**Рисунок 1. График унимодальной функции f(x) = cos(x)th(x)**



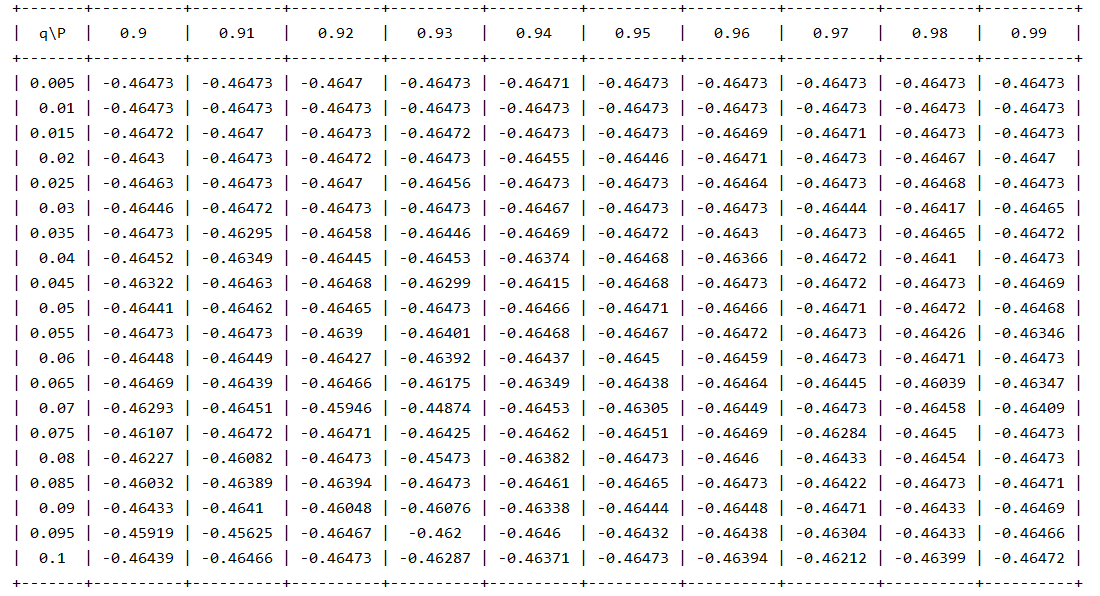
**Рисунок 2. График мультимодальной функции f(x) = cos(x)th(x)\*sin(5x)**

При выполнении лабораторной работы было важно соблюсти указанную последовательность действий, а именно:

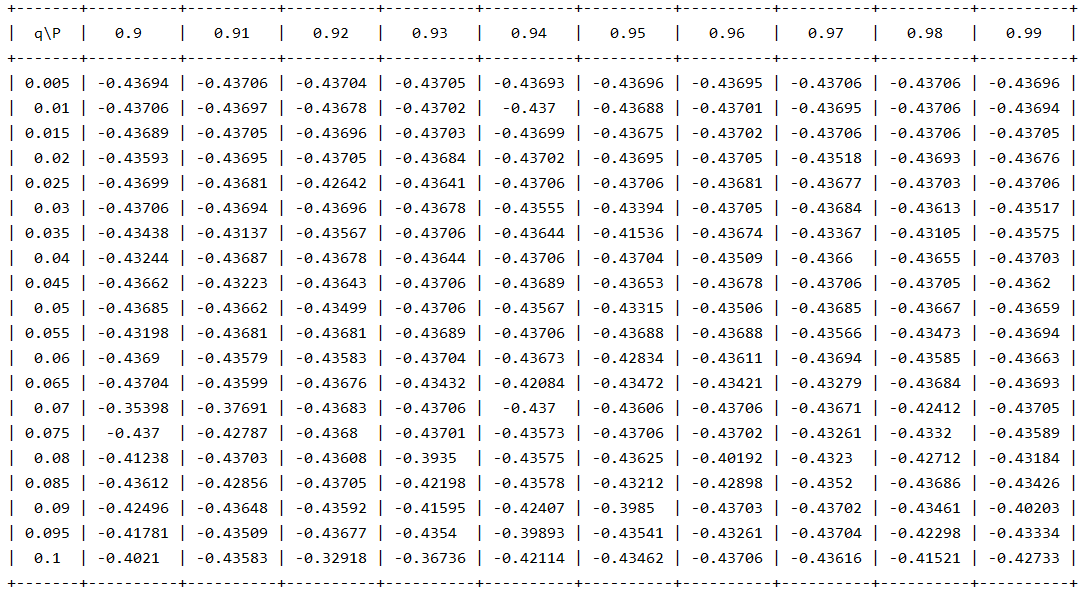
* Определение вероятности непопадания в -окрестность экстремума за одно испытание;
* записать выражение для вероятности непопадания в ε -окрестность экстремума за испытаний;
* из выражения для определить необходимое число испытаний в зависимости от заданных и ε .

Программа сформировала следующие таблицы для унимодальной и мультимодальной функций соответственно.

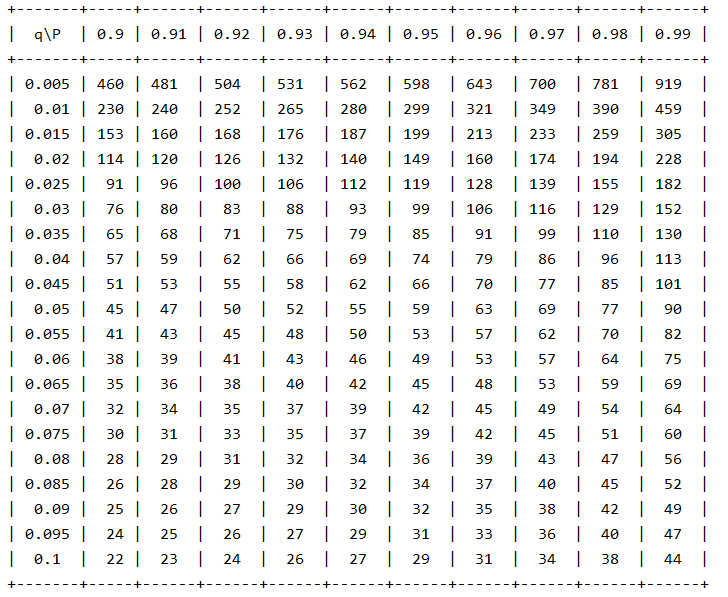
**Таблица 1. Результаты поиска минимума f(x) в зависимости от P и q**



**Таблица 2. Результаты поиска минимума f(x)\*sin(5x) в зависимости от P и q**



**Таблица 3. Зависимость N от P и q**



# **Вывод**

Особенность изученного метода случайного поиска в том, что для случайные испытания строят совершенно не зависимо от результатов предыдущих. Сходимость такого поиска очень мала, но имеется важное преимущество: возможность решать многоэкстремальные задачи (искать глобальный экстремум). Для унимодальной и мультимодальной функций алгоритм случайного поиска выполняется одинаково эффективно.

# **Листинг**

*Прим. Представленный код написан на Python с помощью среды PyCharm.*

**from** math **import** cos, tanh, ceil, log, sin *# Импортируем субмодули математических значений***from** random **import** uniform *# Импортируем субмодуль для генерирования чисел согласно равномерному распределению***from** numpy **import** arange *# Импортируем субмодуль создания массива***from** prettytable **import** PrettyTable *# Импортируем субмодуль создания простых таблиц ASCII***from** matplotlib **import** pyplot **as** plt *# Импортируем субмодуль для построения графиков  
  
# Таблица зависимости N от P и q (опционально)***def** Dependent():  
 table = PrettyTable() *# Создание таблицы* table.add\_column(**"q\\P"**, list(map(**lambda** N: round(N, 3), arange(0.005, 0.105, 0.005)))) *# Формируем столбцы таблицы* **for** p **in** arange(0.9, 1, 0.01): *# Запускаем цикл:* column = []  
 **for** q **in** arange(0.005, 0.105, 0.005):  
 N = ceil(log(1 - p) / log(1 - q)) *# Формула вычисления количества случайных точек* column.append(N)  
 table.add\_column(str(round(p, 2)), column)  
 print(table)  
  
*# Реализация метода случайного поиска***def** RandomSearchMethod(f, a, b):  
 table = PrettyTable() *# Создание таблицы* table.add\_column(**"q\\P"**, list(map(**lambda** x: round(x, 3), arange(0.005, 0.105, 0.005)))) *# Формируем столбцы таблицы* **for** p **in** arange(0.9, 1, 0.01): *# Запускаем цикл для каждого столбца:* column = []  
 **for** q **in** arange(0.005, 0.105, 0.005):  
 N = ceil(log(1 - p) / log(1 - q)) *# Формула вычисления количества случайных точек* x\_value = [uniform(a, b) **for** i **in** range(N)] *# Возвращает случайное вещественное число в [a,b]* y\_value = list(map(f, x\_value))  
 cur\_min = round(min(y\_value), 5)  
 column.append(cur\_min)  
 table.add\_column(str(round(p, 2)), column)  
 print(table)  
 print(**'Необходимое число попаданий:'**, N)  
  
  
**if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 UnimodalFunction = **lambda** x: cos(x) \* tanh(x) *# Унимодальная функция* MultimodalFunction = **lambda** x: UnimodalFunction(x) \* sin(5 \* x) *# Мультимодальная функция* xBegin = -2.0  
 xEnd = 0.0  
 delta = 0.001  
 xlist = arange(xBegin, xEnd, delta)  
 ylistUni = [UnimodalFunction(x) **for** x **in** xlist]  
 ylistMulti = [MultimodalFunction(x) **for** x **in** xlist]  
  
 *# Построим графики для каждой функции* plt.plot(xlist,ylistUni)  
 plt.title(**"Унимодальная функция"**)  
 plt.grid(**True**)  
 plt.show()  
 plt.plot(xlist, ylistMulti)  
 plt.title(**"Мультимодальная функция"**)  
 plt.grid(**True**)  
 plt.show()  
  
 Dependent()  
 RandomSearchMethod(UnimodalFunction, -2, 0)  
 RandomSearchMethod(MultimodalFunction, -2, 0)