Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**Отчёт**

**по лабораторной работе №1**

**Дисциплина: методы поисковой оптимизации**

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. А. Курбатский

Направление подготовки 02.03.02 Фундаментальная информатика и

информационные технологии

Направленность (профиль) Математическое и программное обеспечение

компьютерных технологий

Преподаватель

д-р техн. наук, проф. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. Е. Полупанова

**Цель работы:** изучить методы безусловной поисковой оптимизации с использованием производных, и применить один из них.

**Ход работы:**

Для реализации был выбран алгоритм градиентного спуска с постоянным шагом.

**Шаги алгоритма:**

*Шаг 1*. Задать *х ,* 0 < ε < 1, ε 1 > 0, ε 2 > 0, *М* –предельное число итераций. Найти градиент функции в произвольной точке .

*Шаг 2.* Положить *k* = 0.

*Шаг 3.* Вычислить антиградиент функции *f(xk).*

*Шаг 4.* Проверить выполнение критерия окончания *|f(x\*)| <* ε1*:*

а) если критерий выполнен, то расчёт закончен и *х\* = xk*;

б) если критерий не выполнен, то перейти к шагу 5.

*Шаг 5.* Проверить выполнение неравенства *k ≥ M:*

а) если неравенство выполнено, то расчет окончен: *х\* = xk*;

б) если нет, то перейти к шагу 6.

*Шаг 6*. Задать величину шага *tk.*

*Шаг 7*. Вычислить *xk+1 = xk - tkf(xk)*.

*Шаг 8*. Проверить выполнение условия

*f*(*xk+1*) *- f*(*xk*) *< 0* (или *|f*(*xk+1*) *- f*(*xk*) *|<*  ε *||f(xk)||2*);

а) если условие выполнено, то перейти к шагу 9;

б) если условие не выполнено, положить  иперейти к шагу 7.  
*Шаг 9*. Проверить выполнение условий

*||xk+1 - xk|| <* ε2*, ||f(xk+1) - f(xk)|| <* ε2:

а) если оба условия выполнены при текущем значении *k* и *k = k -*1, то расчет окончен и *x\* = xk+1*;

б) если хотя бы одно из условий не выполнено, положить *k = k* +1 и перейти к шагу 3.

Для создания программы используется язык программирования Python 3.10 и среда разработки PyCharm. Для графической визуализации используется графический фреймворк Tkinter и библиотека Matplotlib.

В созданной программе одно главное активное окно.

Изображение выглядит как текст, снимок экрана, диаграмма, План

Автоматически созданное описание

Рисунок 1 – Главное окно программы.

В данном окне можно выбрать необходимую лабораторную работу, в данном случае «1», ввести координаты начальной точки, задать шаг алгоритма, максимальное число итераций алгоритма и задержку отображения изменений на графике.

В поле «Выполнение и результаты» динамически выводятся результаты работы алгоритма в виде шагов, представленных координатами и значениями оптимизируемой функции в этих координатах.

В меню «Функция и отображение её графика» можно настроить интервал допустимых значений для параметров X и Y, коэффициент уменьшения значений по оси Z, а также выбрать функцию.

При нажатии на кнопку «Выполнить» на главном окне программы отображается выбранная нами функция, как показано на рисунке 2. Причем, более высокие значения функции показана ярко красным цветом, а самые низкие темно синим цветом.

Изображение выглядит как текст, снимок экрана, диаграмма, План

Автоматически созданное описание

Рисунок 2 – Функция Химмельблау.

Для запуска алгоритма необходимо выбрать функцию и нажать кнопку «Выполнить», при этом поле «Выполнение и результаты» начнёт динамически заполняться, а на отображаемой функции можно наглядно увидеть функционирование алгоритма – проследив за передвижением точки, после окончания работы алгоритма точка загорается красным цветом.

**Вывод:** в ходе работы были изучены различные методы безусловной поисковой оптимизации с использованием производных, реализован метод градиентного спуска с постоянным шагом.