Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Кафедра «Информатика»
кафедра

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 11

Метод Гаусса-Зейделя
Тема

Преподаватель			В. В. Тынченко
		Подпись, дата	Инициалы, Фамилия
Студент	КИ19-17/1Б, №031939174		А. К. Никитин
	Номер группы, зачетной книжки	Подпись, дата	Инициалы, Фамилия

Красноярск 2021

1 Постановка задачи

Разработать программу, реализующую метод наискорейшего градиентного спуска.

Найти безусловный экстремум функции, выбранной в соответствии с заданием, с использованием разработанной программы.

Функция: $(x_2 + x_1 - 1)^2 + 2(x_1 - 2)^2 \to min$

2 Описание метода

Стратегия метода Гаусса-Зейделя состоит в построении последовательности точек $\{x_k\}$, k = 0, 1, ..., таких, что $f(x_k + 1) < f(x_k)$, k = 0, l , ...

Точки последовательности $\{x_k\}$ вычисляются по правилу

$$x^{jk+1} = x^{jk} - t_k \left(\frac{\partial f(x)}{\partial x_{k+1}} \right)_{x=x^{jk}} \cdot e_{k+1}, \qquad (6.6)$$

где j — номер цикла вычислений, j = 0, 1, 2, ...; k — номер итерации внутри цикла, k = 0, 1, ..., n – 1; e_{k+1} — единичный вектор, (k + 1)-я проекция которого равна 1; точка x^{00} задается пользователем, величина шага t_k выбирается из условия

$$\varphi(t_k) = f\left(x^{jk} - t_k \left(\frac{\partial f(x)}{\partial x_{k+1}} \right)_{x=x^{jk}} \cdot e_{k+1} \right) \to \min_{t_k}.$$

Эта задача является задачей одномерной минимизации функции $\varphi(t_k) = f\left(x^{jk} - t_k \left(\frac{\partial f(x)}{\partial x_{k+1}} \right)_{x=x^{jk}} \cdot e_{k+1} \right)$ и может быть решена либо с использованием условий $\frac{d\varphi}{dt_k} = 0, \frac{d^2\varphi}{dt_k^{\,2}} > 0,$ либо численно с использованием методов одномерной минимизации, как задача $\varphi(t_k) \to \min_{t_k \in [a,b]}$.

Если уравнение $\frac{d\varphi}{dt_k} = 0$ имеет высокую степень и корни его трудно определить, можно аппроксимировать функцию $\varphi(t_k)$ полиномом $P(t_k)$ второй или третьей степени и определить t^*_k из условий $\frac{dP}{dt_k} = 0, \frac{d^2P}{dt_k^{\ 2}} > 0.$

При численном решении задачи определения величины шага степень близости найденного значения t_k к оптимальному значению t^*_k, удовлетворяющему условиям $\frac{d\varphi}{dt_k} = 0, \frac{d^2\varphi}{dt_k^{\ 2}} > 0,$ зависит от задания интервала [a, b] и точности методов одномерной минимизации.

Легко видеть, что при фиксированном j за одну итерацию с номером k изменяется только одна проекция точки x^{jk}, имеющая номер $k + 1$, а в течение всего цикла с номером j, т. е. начиная с $k = 0$ и кончая $k = n - 1$, изменяются все n проекций точки x^{j0}. После этого точке x^{jn} присваивается номер $x^{j+1,0}$ и она берется за начальную точку для вычислений в $(j + 1)$-м цикле.

Расчет заканчивается в точке x^{jk} при выполнении по крайней мере одного из трех критериев окончания счета: $\|\nabla f(x^{jk})\| < \varepsilon_1$, или $k \geq M$, или двукратного выполнения неравенств $\left\|x^{jk+1} - x^{jk}\right\| < \varepsilon_2$, $\left|f\left(x^{jk+1}\right) - f\left(x^{jk}\right)\right| < \varepsilon_2$. Здесь ε_1, ε_2— малые положительные числа, M — предельное число циклов итераций.

3 Исходные тексты программ

На листинге 1 представлен код программы, реализующий задание.

Листинг 1 – Метод градиентного спуска с постоянным шагом

```
import numpy as np
from const import func
from sympy import Symbol
from sympy.solvers import solve

t = Symbol('t')
```

Окончание листинга 1

```
def calculate_step(f, x, diff_val, basis):
    t_function = f.calc(x - t * diff_val * basis)
    return float(solve(t_function.diff(t), t)[0])



def generate_n_basis_vectors(n):
    return np.array([[1. if k == j else 0. for k in range(n)] for j in range(n)])



def gauss_seidel_method(f, x0, epsilon1=0.1, epsilon2=0.1, M=100):
    x_list = [[np.array(x0).astype(float)]]
    basis_vectors = generate_n_basis_vectors(len(f.args))
    for j in range(M):
        for k, arg in enumerate(f.args):

            gradient = f.gradient_value(x_list[j][k])
            if np.linalg.norm(gradient) < epsilon1:
                return x_list[j][k], k + 1

            partial_diff_val = f.partial_diff(arg, x_list[j][k])
            step    =    calculate_step(f,    x_list[j][k],    partial_diff_val,
basis_vectors[k])
            x_list[j].append(x_list[j][k]   -   step   *   partial_diff_val   *
basis_vectors[k])

            if np.linalg.norm(x_list[j][k + 1] - x_list[j][k]) < epsilon2 \
                    and abs(f.calc(x_list[j][k + 1]) - f.calc(x_list[j][k])) <
epsilon2 \
                    and len(x_list) > 2 \
                    and np.linalg.norm(x_list[j][k] - x_list[j][k - 1]) < epsilon2
\
                    and abs(f.calc(x_list[j][k]) - f.calc(x_list[j][k - 1])) <
epsilon2:
                return x_list[j][k + 1], k + 1

        x_list.append([x_list[-1][-1]])

    return x_list[-1][-1], M
```

4 Исследование влияния параметров метода на точность и скорость нахождения решения

В качестве гиперпараметров по умолчанию использовались следующие значения:

а) x0 = (-10, 10);

б) $\varepsilon_1 = 0.1$;

в) $\varepsilon_2 = 0.1$;

г) M = 100.

Из рисунка 1 можно заключить, что, в целом, чем меньше параметр ε_1, тем точнее решение, однако тем больше шагов необходимо алгоритму для нахождения ответа.

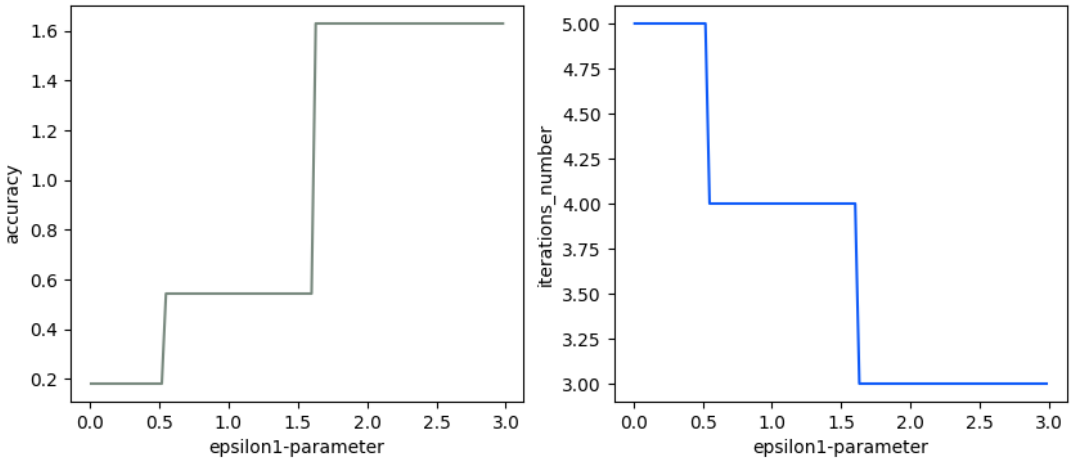

Рисунок 1 – Влияние параметра ε_1 на точность и производительность

Из рисунка 2 можно провести аналогичные рассуждения касательно параметра ε_2.

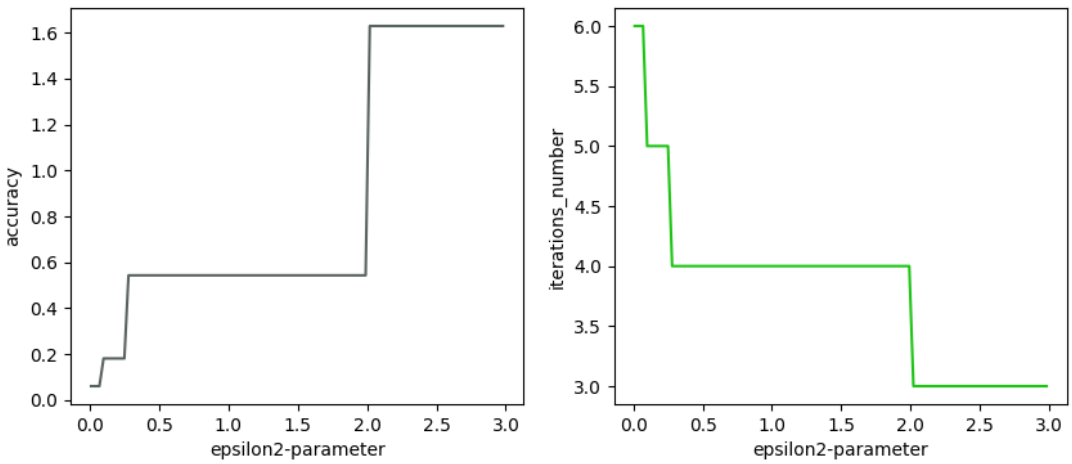

Рисунок 2 - Влияние параметра ε_2 на точность и производительность

Из рисунка 3 можно заключить, что параметр максимального количества шагов лучше не делать слишком небольшим, и что чем он больше, тем точнее, но менее производительнее, программа.

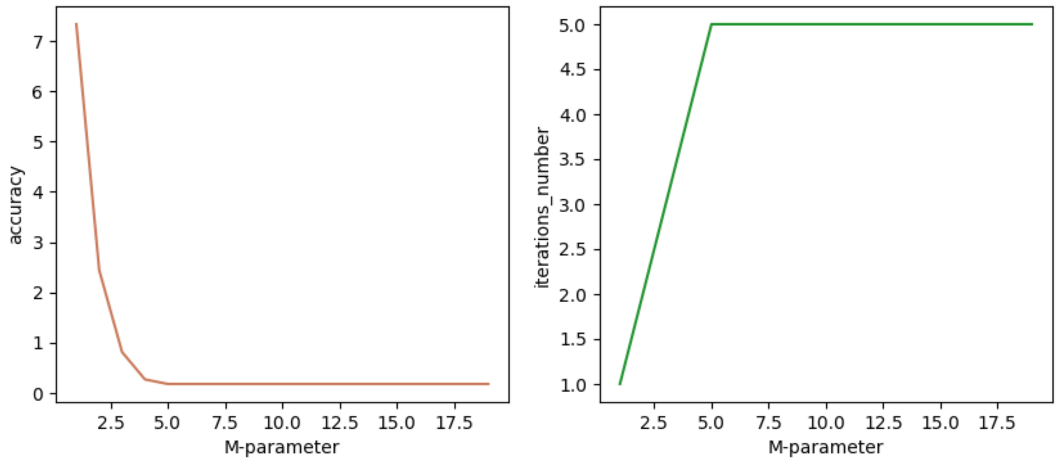

Рисунок 3 - Влияние параметра максимального количества шагов на точность и производительность

5 Вывод

В результате данной работы был реализован и проанализирован метод Гаусса-Зейделя для поиска локального минимума многомерной функции. Также были проанализированы гиперпараметры метода и их влияние на работу функции.