МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА» (БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**Лабораторная работа № 7**по дисциплине: «Вычислительная математика»

Выполнил: ст. группы ПВ-211

Медведев Д.С.

Проверила:

Бондаренко Т.В.

Белгород 2023 г.

Минимизация функции многих переменных методом градиента с дроблением шага

# Вариант 8

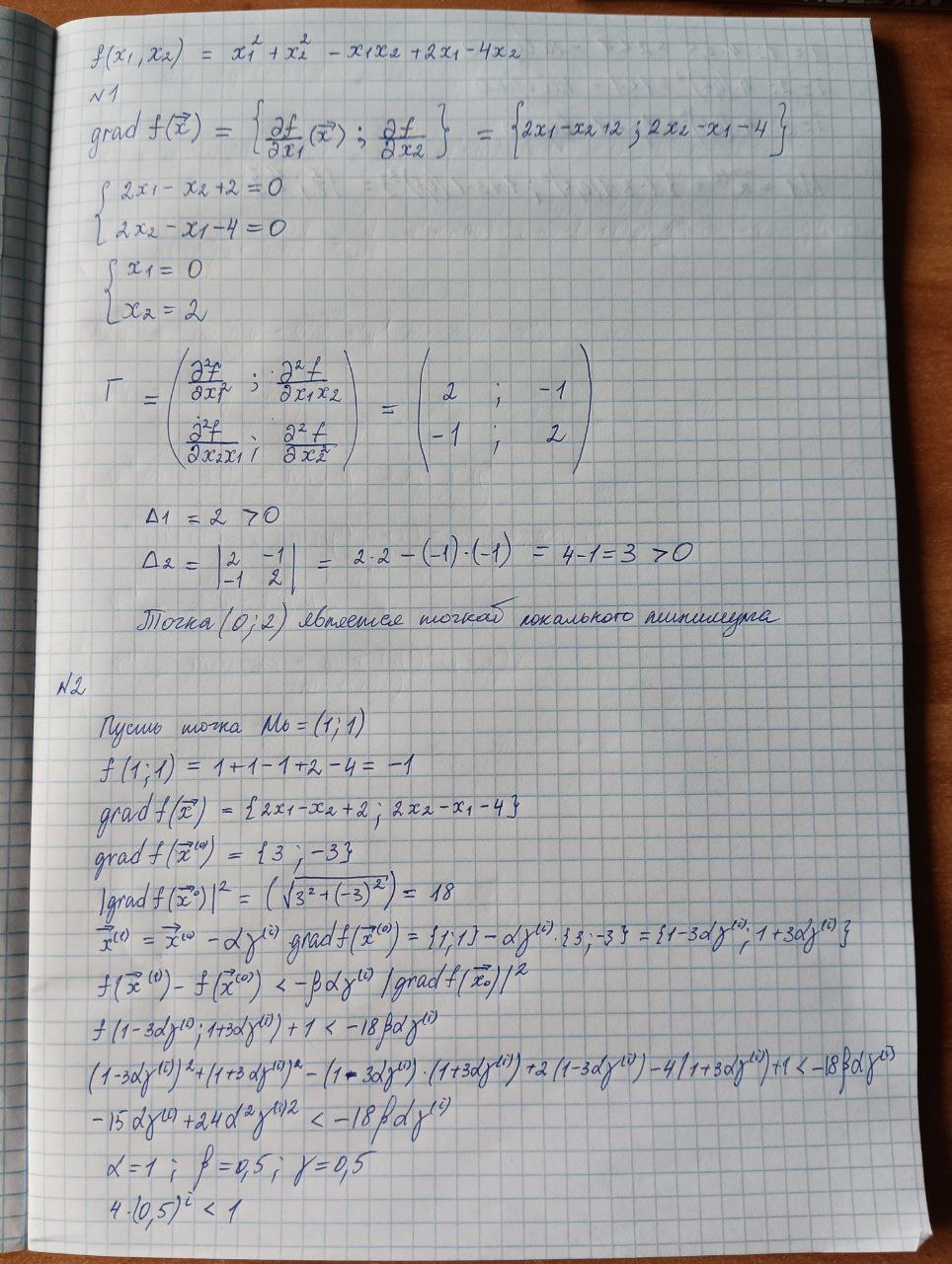
**Цель работы:** изучить метод градиента с дроблением шага для решения задачи минимизации функции многих переменных и получить практические навыки его применения.

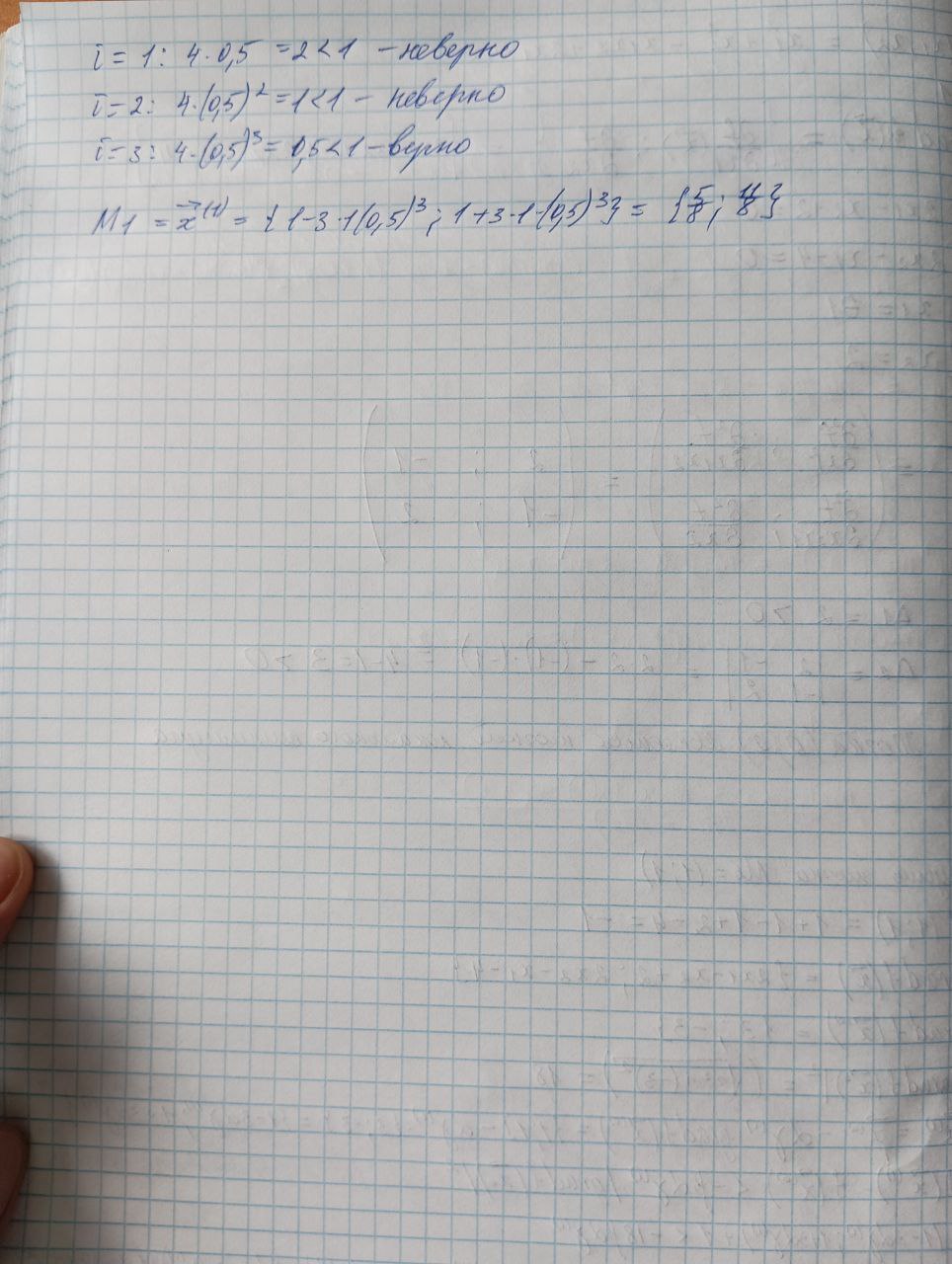
**Ход работы**

****

1. Найти точные значения координат точки минимума и минимальное значение функции у = f (x1, х2) для функции соответствующего варианта задания, используя необходимые и достаточные условия локального минимума.

2. Выполнить вручную вычисление приближенного значения точки минимума целевой функции методом градиента с дроблением шага, начиная с произвольно выбранного начального приближения М0 (x10, x20). Точность решения ε = 0,01. Вручную подробно достаточно выполнить первый шаг метода градиента с дроблением шага. Параметры метода выбрать самостоятельно.







3. Реализовать логическую функцию для нахождения приближенного значения точки локального минимума и минимального значения целевой функции у = f(x1, х2) методом градиента с дроблением шага.

Входными данными для логической функции являются:

* целевая функция у = f (x1, х2);
* градиент целевой функции grad (f (x1, х2));
* начальное приближение М0(x10, x20) к точке локального минимума;
* точность решения ɛ;
* ограничение на максимальное число итераций n;
* параметры метода градиента с дроблением шага α, β, γ.

Функция возвращает значение «истина», если приближенное

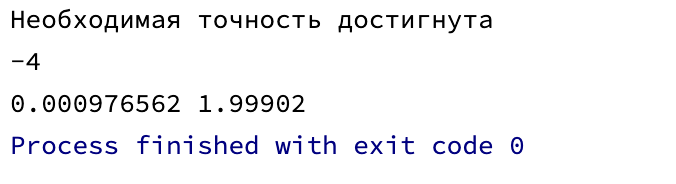
решение с заданной точностью получено за число итераций, не превышающее n, и «ложь» в противном случае.

Результат работы программы: приближенное значение точки локального минимума и минимальное значение целевой функции у = f (x1, х2) с заданной точностью, количество выполненных итераций.

Предусмотреть возможность сохранения пошаговых результатов реализации метода градиента с дроблением шага в файл.

#include **<iostream>**#include **"cmath"  
  
using namespace** std;  
  
**struct** Point {  
 **float** x1;  
 **float** x2;  
};  
  
**typedef float** func(Point point);  
  
**typedef** Point grad(Point point);  
  
**float** targetFunction(Point point) {  
 **return** pow(point.x1, 2) + pow(point.x2, 2) - point.x1 \* point.x2 + 2 \* point.x1 - 4 \* point.x2;  
}  
  
Point gradient(Point point) {  
 **return** {2 \* point.x1 - point.x2 + 2, 2 \* point.x2 - point.x1 - 4};  
}  
  
**bool** findMinPoint(func targetFunction, grad gradient, Point &point, **float** &minValue, **float** eps, **int** maxIterations,  
 **float** alpha, **float** betta, **float** gamma) {  
 **float** precision = eps \* 2;  
 **int** iteration = 0;  
  
 **while** (precision > eps && iteration < maxIterations) {  
 Point currentGradient = gradient(point);  
 **float** gradientModuleSquared = pow(currentGradient.x1, 2) + pow(currentGradient.x2, 2);  
 Point newPoint;  
  
 **int** i = 0;  
 **bool** equationIsValid = **false**;  
 **while** (!equationIsValid) {  
 i++;  
 newPoint = {(**float**) (point.x1 - alpha \* pow(gamma, i) \* currentGradient.x1),  
 (**float**) (point.x2 - alpha \* pow(gamma, i) \* currentGradient.x2)};  
  
 equationIsValid = targetFunction(newPoint) - targetFunction(point) <  
 -betta \* alpha \* pow(gamma, i) \* gradientModuleSquared;  
 }  
  
 precision = abs(targetFunction(newPoint) - targetFunction(point));  
 point = newPoint;  
 iteration++;  
 }  
  
 minValue = targetFunction(point);  
 **return** precision < eps;  
}  
  
  
**int** main() {  
 Point point = {1, 1};  
 **float** minValue;  
 **bool** isSuccessful = findMinPoint(targetFunction, gradient, point, minValue, 0.0001, 100, 1, 0.5, 0.5);  
  
 **if** (isSuccessful) {  
 cout << **"Необходимая точность достигнута"**;  
 } **else** {  
 cout << **"Необходимая точность не достигнута"**;  
 }  
  
 cout << **"\n"** << minValue << **"\n"** << point.x1 << **" "** << point.x2;  
}

**Пример вывода программы:**

****

**Вывод:** в ходе лабораторной работы мы изучили метод градиента с дроблением шага для решения задачи минимизации функции многих переменных и получили практические навыки его применения.