Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Российский химико-технологический университет имени Д.И. Менделеева»

Факультет цифровых технологий и химического инжиниринга

Кафедра информационных компьютерных технологий

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 11**

**ПО КУРСУ**

**«ЧИСЛЕННЫЕ МЕТОДЫ В СРЕДЕ MATLAB»:**

**«Оптимизация»**

Ведущий преподаватель

Доцент кафедры ИКТ Филиппова Е.Б.

**СТУДЕНТ группы КС-20** Мелехин А.А.

**Москва**

**2024**

# **Задание (Вариант 14)**

**Задание 1:** Для производства метанола компании Химио необходимо купить цилиндрическую емкость объемом 1200 м3. При этом им необходимо минимизировать тепловые потери будущей емкости. Тепловой поток может быть разным через отдельные поверхности емкости. Это объяснено наличием ветра около боковых стенок емкости, что приводит к увеличению коэффициента теплоотдачи от стенок бака к воздуху. Будем полагать, что теплоотдача через единицу поверхности дна бака равна 0.35 от теплоотдачи через единицу поверхности стенки бака. Для крышки бака данное значение равно 0.68. Определите оптимальное соотношение диаметра бака к его высоте для минимизации тепловых потерь.

При решении задачи исходить из того, что площадь крышки и дна бака вычисляется по формуле:

Площадь боковой поверхности:

А объем емкости:

Необходимо найти оптимальное решение методами Золотого сечения, парабол и Ньютона и сравнить результаты.

**Задание 2:** при помощи метода градиентного спуска необходимо найти минимум данной функции 2-х переменных в диапазоне поиска по каждой координате [-1.5; 4] с точностью 10-4:

**Код (Программа lab11.m)**

clc; clear;

% Задание 1

% Начальный интервал

a = 10;

b = 30;

d0 = 5; % Начальное приближение

tol = 1e-4; % Точность и максимальное количество итераций

max\_iter = 100;

% Вызов методов Ньютона, Золотого чесения и парабол

min\_d\_newton = newton\_method(d0, tol, max\_iter);

min\_d\_golden = golden\_section\_method(a, b, tol, max\_iter);

min\_d\_parabolic = parabolic\_method(a, b, tol, max\_iter);

% Вызов функции MATLAB

options = optimset('Display', 'off');

[min\_d\_matlab, min\_heat\_loss] = fminsearch(@calculate\_heat\_loss, d0, options);

% Вывод результата

disp('----------------------------------------------------------------------------------')

disp('Задание 1')

disp(['Оптимальное значение d (метод Ньютона): ', num2str(min\_d\_newton)]);

disp(['Оптимальное значение d (метод золотого сечения): ', num2str(min\_d\_golden)]);

disp(['Оптимальное значение d (метод парабол): ', num2str(min\_d\_parabolic)]);

disp(['Оптимальное значение d (стандартная функция MATLAB): ', num2str(min\_d\_matlab)]);

disp('----------------------------------------------------------------------------------')

% Задание 2

f = @(x) sin(x(1) + x(2)) + (x(1) - x(2)).^2 - 1.5\*x(1) + 2.5\*x(2) + 1; % Функция для минимизации

% Градиент функции f

grad\_f = @(x) [cos(x(1) + x(2)) + 2\*(x(1) - x(2)) - 1.5;

cos(x(1) + x(2)) - 2\*(x(1) - x(2)) + 2.5];

x0 = [0; 0]; % Начальное приближение

% Параметры метода

alpha = 0.01;

tol = 1e-4;

max\_iter = 1000;

% Метод градиентного спуска

[xmin\_custom, fmin\_custom, steps\_custom] = gradient\_descent(f, grad\_f, x0, alpha, tol, max\_iter);

% Стандартный метод MATLAB

xmin\_builtin = fminsearch(f, x0);

fmin\_builtin = f(xmin\_builtin);

% Вывод результатов

disp(' ')

disp('----------------------------------------------------------------------------------')

disp('Задание 2')

disp('Метод градиентного спуска:');

disp(['Минимум: ', num2str(fmin\_custom)]);

disp(['Точка минимума: (', num2str(xmin\_custom(1)), ', ', num2str(xmin\_custom(2)), ')']);

disp('Стандартный метод MATLAB:');

disp(['Минимум: ', num2str(fmin\_builtin)]);

disp(['Точка минимума: (', num2str(xmin\_builtin(1)), ', ', num2str(xmin\_builtin(2)), ')']);

disp('----------------------------------------------------------------------------------')

% График значений функции на отдельных шагах поиска экстремума

figure;

plot(1:length(steps\_custom), steps\_custom, 'bo-', 'LineWidth', 1.5);

hold on;

xlabel('Шаги');

ylabel('f(x,y)');

title('Значения функции на шагах градиентного спуска');

grid on;

% График сам найденного экстремума (собственный метод)

plot(length(steps\_custom), fmin\_custom, 'r\*', 'LineWidth', 2);

% График сам найденного экстремума (fminsearch)

plot(length(steps\_custom), f(xmin\_builtin), 'g\*', 'LineWidth', 2);

legend('Значения функции на шагах', 'Найденный минимум (метод градиентного спуска)', 'Найденный минимум (Стандартный метод MATLAB)');

hold off;

function heat\_loss = calculate\_heat\_loss(d) % Функция для вычисления потерь тепла

h = (4\*1200)/(pi\*d.^2); % выражение высоты исходя из общего объёма

% Коэффициенты теплоотдачи

h\_bottom = 0.35; % Для дна

h\_wall = 1; % Для стенок

h\_lid = 0.68; % Для крышки

% Площади поверхностей

S\_bottom = pi \* (d^2) / 4;

S\_wall = pi \* d \* h;

S\_lid = S\_bottom;

% Тепловые потери через каждую поверхность

Q\_bottom = h\_bottom \* S\_bottom;

Q\_wall = h\_wall \* S\_wall;

Q\_lid = h\_lid \* S\_lid;

% Общие тепловые потери

heat\_loss = Q\_bottom + Q\_wall + Q\_lid;

end

function min\_d = newton\_method(d0, tol, max\_iter) % метод Ньютона

d = d0;

iter = 0;

while iter < max\_iter

h\_loss = calculate\_heat\_loss(d);

h = 1e-4; % малое приращение для численного дифференцирования

d\_prime = (calculate\_heat\_loss(d + h) - calculate\_heat\_loss(d)) / h;

d\_double\_prime = (calculate\_heat\_loss(d + h) - 2 \* h\_loss + calculate\_heat\_loss(d - h)) / h^2;

d = d - d\_prime / d\_double\_prime;

if abs(d\_prime) < tol

break;

end

iter = iter + 1;

end

min\_d = d;

end

function min\_d = golden\_section\_method(a, b, tol, max\_iter) % метод золотого сечения

rho = (sqrt(5) - 1) / 2;

d = a + rho \* (b - a);

c = b - rho \* (b - a);

iter = 0; % Инициализация переменной iter

while abs(b - a) > tol && iter < max\_iter

if calculate\_heat\_loss(c) < calculate\_heat\_loss(d)

b = d;

d = c;

c = b - rho \* (b - a);

else

a = c;

c = d;

d = a + rho \* (b - a);

end

iter = iter + 1; % Увеличение счетчика итераций

end

min\_d = (a + b) / 2;

end

function min\_d = parabolic\_method(a, b, tol, max\_iter) % метод парабол

x = [a, (a + b) / 2, b];

f = zeros(1, 3);

f(1) = calculate\_heat\_loss(x(1));

f(2) = calculate\_heat\_loss(x(2));

f(3) = calculate\_heat\_loss(x(3));

iter = 0;

while abs(b - a) > tol && iter < max\_iter

A = ((x(2) - x(3)) \* f(1) + (x(3) - x(1)) \* f(2) + (x(1) - x(2)) \* f(3)) / ...

((x(1) - x(2)) \* (x(2) - x(3)) \* (x(3) - x(1)));

B = (f(2) - f(1)) / (x(2) - x(1)) - (x(1) + x(2)) \* A;

C = f(1) - A \* x(1)^2 - B \* x(1);

min\_x = -B / (2 \* A);

if min\_x < x(2)

if calculate\_heat\_loss(min\_x) < f(2)

x(3) = x(2);

x(2) = min\_x;

f(3) = f(2);

f(2) = calculate\_heat\_loss(min\_x);

else

x(1) = min\_x;

f(1) = calculate\_heat\_loss(min\_x);

end

else

if calculate\_heat\_loss(min\_x) < f(2)

x(1) = x(2);

x(2) = min\_x;

f(1) = f(2);

f(2) = calculate\_heat\_loss(min\_x);

else

x(3) = min\_x;

f(3) = calculate\_heat\_loss(min\_x);

end

end

iter = iter + 1;

end

min\_d = x(2);

end

function [xmin, fmin, steps] = gradient\_descent(f, grad\_f, x0, alpha, tol, max\_iter) % Метод градиентного спуска

% Градиентный спуск для минимизации функции f

% f - функция для минимизации

% grad\_f - градиент функции f

% x0 - начальное приближение

% alpha - скорость обучения

% tol - допустимая ошибка

% max\_iter - максимальное количество итераций

x = x0;

steps = [];

for iter = 1:max\_iter

grad = grad\_f(x);

x = x - alpha \* grad;

steps = [steps; f(x)];

% Проверка на сходимость

if norm(grad) < tol

break;

end

end

xmin = x;

fmin = f(x);

end

**Результаты расчётов**

----------------------------------------------------------------------------------

Задание 1

Оптимальное значение d (метод Ньютона): 14.369

Оптимальное значение d (метод золотого сечения): 14.3691

Оптимальное значение d (метод парабол): 14.3566

Оптимальное значение d (стандартная функция MATLAB): 14.3691

----------------------------------------------------------------------------------

----------------------------------------------------------------------------------

Задание 2

Метод градиентного спуска:

Минимум: -1.9132

Точка минимума: (-0.54716, -1.5472)

Стандартный метод MATLAB:

Минимум: -1.9132

Точка минимума: (-0.54721, -1.5472)

----------------------------------------------------------------------------------