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 № 14**

**ПО КУРСУ**

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

**«Нелинейное уравнение»**

Ведущий преподаватель

Доцент кафедры ИКТ Филиппова Е.Б.

**СТУДЕНТ группы КС-20** Мелехин А.А.

**Москва**

**2024**

# **Задание**

**№14 Про кольцо Марволо Мракса**: прошло уже 15 с половиной лет, когда Темный Лорд пришел в Годрикову впадину и пал, пытаясь погубить Гарри Поттера. Но сейчас он снова вернулся, и профессор Дамблдор ищет способы победить Темного Лорда. До него доходят слухи об одном кольце, принадлежавшем когда-то отцу Тома Реддла. Он трансгрессирует в места, где когда-то жила его семья, но никак не может найти дом Реддлов. Один из местных соглашается показать, где находится усадьба, но прежде решает проверить, достоин ли его спутник и выдержит ли он то, что таит это страшное место. В качестве испытания он дает следующее уравнение:

Решите уравнение и помогите Дамблдору остановить «Того-Кого-Нельзя-Называть».

Необходимо локализовать корень (корни) методом перебора или половинного деления, а затем уточнить корень (корни) методами простых итераций, хорд, и секущих.

**Программа lab5.m**

clear; clc; % очистка командного окна и переменных

% Определение функции для решаемого уравнения

f = @(x) -3 \* (x .^ 3) + 4 \* x + 5;

% Определение интервала для перебора корней

a = 1; % начало интервала

b = 2; % конец интервала

n = 1000; % количество точек для перебора

x0 = 1; % Начальное приближение

x\_fzero = fzero(f, x0);

x\_fsolve = fsolve(f, x0);

fprintf('-------------------------\n');

disp(['fzero solution: x = ', num2str(x\_fzero)]);

disp(['fsolve solution: x = ', num2str(x\_fsolve)]);

fprintf('-------------------------\n\n');

% Метод перебора

x = linspace(a, b, n); % генерация n равномерно распределенных точек на интервале [a, b]

y = f(x); % вычисление значений функции в этих точках

roots = x(abs(y) < 0.01); % выбор корней с определенной точностью

% Результаты перебора

fprintf('-------------------------\n');

fprintf('Метод перебора:\n');

disp(['Корни уравнения: ', num2str(roots)]);

fprintf('-------------------------\n\n');

% График функции и найденных корней

figure;

plot(x, y, 'b'); % построение графика функции

hold on;

plot(roots, f(roots), 'm\*'); % отметка корней на графике

xlabel('x');

ylabel('f(x)');

title('График функции и найденных корней');

legend('Функция f(x)', 'Корни', 'Location', 'NorthWest');

% метод половинного деления

% Интервал начального приближения a и b

a = 1;

b = 2;

tol = 1e-6;

maxIter = 100;

converge = true; % предполагаем, что сходится

if f(a) \* f(b) > 0 % Проверка условия сходимости

converge = false; % не сходится

end

fprintf('-------------------------\n');

fprintf('Метод половинного деления:\n');

if converge

fprintf('Сходимость метода: сходится\n');

c = (a + b) / 2; % Начальное приближение

iter = 0;

while abs(f(c)) > tol && iter < maxIter

if f(a) \* f(c) < 0

b = c;

else

a = c;

end

c = (a + b) / 2;

iter = iter + 1;

end

fprintf('Корень: %f \nЧисло итераций: %d\n', c, iter);

else

fprintf('Сходимость метода: не сходится\n');

end

fprintf('-------------------------\n\n');

% Метод простых итераций

t = 0.1;

g = @(x) x+t\*(-3 \* (x .^ 3) + 4 \* x + 5); % итерационная формула

dg = @(x) -9\*t\*(x.^2)+4\*t+1; % Производная функции g(x)

x = 1.7; % Выбор начального значения x в окрестности корня

maxIter = 100; % Максимальное количество итераций

converge = true; % предполагаем, что сходится

iter = 0;

while abs(dg(x)) >= 1 && iter < maxIter

x = g(x);

iter = iter + 1;

end

if abs(dg(x)) >= 1 % Проверка условия сходимости |g'(x)| < 1 в окрестности корня

converge = false; % не сходится

end

fprintf('-------------------------\n');

fprintf('Метод простых итераций:\n');

if converge

fprintf('Сходимость метода: сходится\n');

iter = 0;

while iter < maxIter

x = g(x0);

if abs(x - x0) < tol

break;

end

x0 = x;

iter = iter + 1;

end

fprintf('Корень: %f\nЧисло итераций: %d\n', x, iter);

else

fprintf('Сходимость метода: не сходится\n');

end

fprintf('-------------------------\n\n');

dg = @(x) 6\*(x.^2)-3;

% метод хорд

% Начальные приближения x0 и x1

x0 = -1;

x1 = 1;

tol = 1e-6; % Точность

maxIter = 100; % Максимальное количество итераций

converge = true; % предполагаем, что сходится

iter = 0;

if dg(x) < 0

converge = false; % не сходится

end

fprintf('-------------------------\n');

fprintf('Метод хорд:\n');

if converge

fprintf('Сходимость метода: сходится\n');

while abs(x1 - x0) >= tol && iter < maxIter

x = x1 - (f(x1) \* (x1 - x0)) / (f(x1) - f(x0));

if abs(x - x1) < tol

break;

end

x0 = x1;

x1 = x;

iter = iter + 1;

end

fprintf('Корень %f\nЧисло итераций: %d\n', x, iter);

else

fprintf('Сходимость метода: не сходится\n');

end

fprintf('-------------------------\n\n');

% метод секущих

% Начальные приближения x0 и x1

x0 = -1;

x1 = 1;

tol = 1e-6; % Точность

maxIter = 100; % Максимальное количество итераций

converge = true; % предполагаем, что сходится

if dg(x) < 0

converge = false; % не сходится

end

fprintf('-------------------------\n');

fprintf('Метод секущих:\n');

if converge

fprintf('Сходимость метода: сходится\n');

iter = 0;

while abs(x1 - x0) >= tol && iter < maxIter

x = x1 - (f(x1) \* (x1 - x0)) / (f(x1) - f(x0));

if abs(x - x1) < tol

break;

end

x0 = x1;

x1 = x;

iter = iter + 1;

end

fprintf('Корень %f\nЧисло итераций: %d\n', x, iter);

else

fprintf('Сходимость метода: не сходится\n');

end

fprintf('-------------------------\n\n');

**Результаты расчётов**

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

fzero solution: x = 1.5516

fsolve solution: x = 1.5516

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

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

Метод перебора:

Корни уравнения: 1.5516

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

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

Метод половинного деления:

Сходимость метода: сходится

Корень: 1.551608

Число итераций: 19

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

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

Метод простых итераций:

Сходимость метода: сходится

Корень: 1.551609

Число итераций: 44

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

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

Метод хорд:

Сходимость метода: сходится

Корень 1.551608

Число итераций: 9

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

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

Метод секущих:

Сходимость метода: сходится

Корень 1.551608

Число итераций: 9

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

