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 ЗАЧЁТНОЙ РАБОТЕ**

**ПО КУРСУ**

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

Ведущий преподаватель

Доцент кафедры ИКТ Филиппова Е.Б.

**СТУДЕНТ группы КС-20** Мелехин А.А.

**Москва**

**2024**

# **Задание**

**1** Решить систему нелинейных алгебраических уравнений **(вариант 6)**:

Начальные приближения: x=1,25 y=2,5

**Программа (task1.m)**

clc; clear;

% Начальные приближения

start = [1.25, 2.5];

% Решение системы уравнений с использованием fsolve

options = optimoptions('fsolve', 'Display', 'iter');

solution = fsolve(@system\_eqs, start, options);

% Вывод решения fsolve

disp('Решение системы уравнений с использованием fsolve:');

disp(['x = ', num2str(solution(1))]);

disp(['y = ', num2str(solution(2))]);

% Начальные приближения для метода Ньютона

x0 = 1.25;

y0 = 2.5;

tol = 1e-6;

max\_iter = 100;

% Проверка сходимости для метода Ньютона

if newton\_convergence\_check(x0, y0)

% Решение методом Ньютона

[x\_newton, y\_newton] = newton\_method(x0, y0, tol, max\_iter);

% Вывод решения методом Ньютона

disp('Решение методом Ньютона:');

disp(['x = ', num2str(x\_newton)]);

disp(['y = ', num2str(y\_newton)]);

else

disp('Метод Ньютона не сходится для данных начальных приближений.');

end

% Начальные приближения для метода простых итераций

x0 = 1.25;

y0 = 2.5;

tol = 1e-6;

max\_iter = 100;

% Проверка сходимости для метода простых итераций

if simple\_iterations\_convergence\_check(x0, y0)

% Решение методом простых итераций

[x\_iter, y\_iter] = simple\_iterations(x0, y0, tol, max\_iter);

% Вывод решения методом простых итераций

disp('Решение методом простых итераций:');

disp(['x = ', num2str(x\_iter)]);

disp(['y = ', num2str(y\_iter)]);

else

disp('Метод простых итераций не сходится для данных начальных приближений.');

end

% Построение графика

fimplicit(@(x, y) sin(x + 1) - y - 1.2, [-2, 2, -2, 2]);

hold on;

fimplicit(@(x, y) 2\*x + cos(y) - 2, [-2, 2, -2, 2]);

% Отметка найденного решения

plot(solution(1), solution(2), 'ro', 'MarkerSize', 10, 'DisplayName', 'fsolve');

if exist('x\_newton', 'var') && exist('y\_newton', 'var')

plot(x\_newton, y\_newton, 'go', 'MarkerSize', 10, 'DisplayName', 'Newton');

end

if exist('x\_iter', 'var') && exist('y\_iter', 'var')

plot(x\_iter, y\_iter, 'bo', 'MarkerSize', 10, 'DisplayName', 'Simple Iterations');

end

% Настройка графика

legend('show');

xlabel('x');

ylabel('y');

title('График уравнений и найденные решения');

grid on;

hold off;

% Определение системы уравнений

function F = system\_eqs(vars)

x = vars(1);

y = vars(2);

F(1) = sin(x + 1) - y - 1.2;

F(2) = 2\*x + cos(y) - 2;

end

function [x, y, converged] = newton\_method(x0, y0, tol, max\_iter)

converged = false;

for iter = 1:max\_iter

% Вычисление значений функций и их производных

F1 = sin(x0 + 1) - y0 - 1.2;

F2 = 2\*x0 + cos(y0) - 2;

J11 = cos(x0 + 1);

J12 = -1;

J21 = 2;

J22 = -sin(y0);

% Якобиан

J = [J11, J12; J21, J22];

% Вектор функций

F = [F1; F2];

% Решение системы линейных уравнений

delta = J \ (-F);

% Обновление значений

x0 = x0 + delta(1);

y0 = y0 + delta(2);

% Проверка условия остановки

if norm(delta) < tol

converged = true;

break;

end

end

x = x0;

y = y0;

end

% Метод простых итераций

function [x, y, converged] = simple\_iterations(x0, y0, tol, max\_iter)

converged = false;

for iter = 1:max\_iter

% Обновление значений

x\_new = (2 - cos(y0)) / 2;

y\_new = sin(x0 + 1) - 1.2;

% Проверка условия остановки

if abs(x\_new - x0) < tol && abs(y\_new - y0) < tol

converged = true;

break;

end

% Обновление переменных

x0 = x\_new;

y0 = y\_new;

end

x = x0;

y = y0;

end

% Проверка сходимости для метода Ньютона

function converged = newton\_convergence\_check(x0, y0)

% Вычисление значений производных

J11 = cos(x0 + 1);

J12 = -1;

J21 = 2;

J22 = -sin(y0);

% Определение Якобиана

J = [J11, J12; J21, J22];

% Проверка невырожденности Якобиана

if det(J) ~= 0

converged = true;

else

converged = false;

end

end

% Проверка сходимости для метода простых итераций

function converged = simple\_iterations\_convergence\_check(x0, y0)

% Вычисление значений производных

g1\_x = 0; % Производная функции g1 по x

g1\_y = sin(y0); % Производная функции g1 по y

g2\_x = cos(x0 + 1); % Производная функции g2 по x

g2\_y = 0; % Производная функции g2 по y

% Определение матрицы Якоби

J = [g1\_x, g1\_y; g2\_x, g2\_y];

% Проверка нормы матрицы Якоби

if norm(J, 'fro') < 1

converged = true;

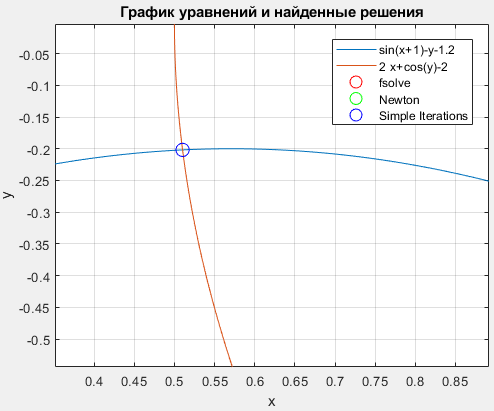
else

converged = false;

end

end

**Результаты расчётов (task1.m)**



Решение системы уравнений с использованием fsolve:

x = 0.51015

y = -0.20184

Решение методом Ньютона:

x = 0.51015

y = -0.20184

Решение методом простых итераций:

x = 0.51015

y = -0.20184