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»:**

**«Системы нелинейных уравнений»**

Ведущий преподаватель

Доцент кафедры ИКТ Филиппова Е.Б.

**СТУДЕНТ группы КС-20** Мелехин А.А.

**Москва**

**2024**

**Задание**

Немногие знают, что самой лучшей медициной в течение многих лет считалась арабская. Уже в средние века арабские медики изобрели катетер и проводили сложнейшие операции, в том числе и на глазах. Основой их знаний составляли науки: математика, физика, химия и др. Так Абу Бакр Мухаммед ар-Рази, смог успешно провести операцию по удалению зуба решив всего одну систему уравнений. Вот она:

Решите систему уравнений и покажите, что Вы тоже разбираетесь в медицине.

Необходимо решить заданную систему нелинейных уравнений методами простых итераций и Ньютона. Перед началом расчёта необходимо построить графики, для того, чтобы определить начальную точку поиска. Делая задания, следует проверять выполнение условий сходимости методов. Фиксировать число требуемых итераций. Сделать проверку, подставив найденные корни в уравнения, а также решить уравнения, используя стандартные операторы MATLAB. Кроме того, необходимо подготовить отчёт, в котором будут отражены блок-схемы и коды алгоритмов, а также приведены результаты работы.

**Программа lab6.m**

clear; clc;

% Заданные уравнения

eq1 = @(x, y) (2 \* x.^2 + 2 \* y.^2 - 1).^5 - 22 \* x.^2 \* y.^3;

eq2 = @(x) -0.05 \* x.^2 - 2.5 \* x + 0.25;

% Начальное приближение для графического поиска

x\_vals = linspace(-2, 2, 400);

y\_vals = eq2(x\_vals);

% Построение графиков

figure;

plot(x\_vals, y\_vals);

hold on;

fimplicit(eq1, [-2, 2, -2, 2], 'b');

xlabel('x');

ylabel('y');

title('Графики функций');

legend('y = -0.05 \* x^2 - 2.5 \* x + 0.25', '(2 \* x^2 + 2 \* y^2 - 1)^5 - 22 \* x^2 \* y^3 = 0');

grid on;

% Начальная точка для итерационных методов (взята из графика)

x0 = -1;

y0 = eq2(x0);

% Метод простых итераций

max\_iter = 1000;

tolerance = 1e-6;

% Настройка параметров для метода Ньютона

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

% Решение системы уравнений методом простых итераций

[x\_pi, y\_pi, iter\_pi] = simple\_iteration\_method(eq1, x0, y0, tolerance, max\_iter);

if isnan(x\_pi) || isnan(y\_pi)

disp('Метод простых итераций не сошелся');

else

disp('Метод простых итераций:');

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

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

disp(['Количество итераций: ', num2str(iter\_pi)]);

end

% Решение системы уравнений методом Ньютона

[x\_newton, y\_newton, iter\_newton] = newton\_method(eq1, x0, y0, tolerance, max\_iter, options);

if isnan(x\_newton) || isnan(y\_newton)

disp('Метод Ньютона не сошелся');

else

disp('Метод Ньютона:');

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

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

disp(['Количество итераций: ', num2str(iter\_newton)]);

end

% Решение с использованием встроенной функции fsolve

eq\_system = @(vars) [eq1(vars(1), vars(2)); eq2(vars(1)) - vars(2)];

[x\_fsolve, ~, exitflag] = fsolve(eq\_system, [x0, y0], options);

if exitflag <= 0

disp('Функция fsolve не смогла найти решение');

else

disp('Функция fsolve:');

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

disp(['y = ', num2str(x\_fsolve(2))]);

end

% Решение методом символьной математики vpasolve

syms x\_sym y\_sym;

eq1\_sym = (2 \* x\_sym^2 + 2 \* y\_sym^2 - 1)^5 - 22 \* x\_sym^2 \* y\_sym^3 == 0;

eq2\_sym = y\_sym == -0.05 \* x\_sym^2 - 2.5 \* x\_sym + 0.25;

[solX, solY] = vpasolve([eq1\_sym, eq2\_sym], [x\_sym, y\_sym]);

disp('Метод символьной математики vpasolve:');

disp('x = ');

disp(solX);

disp('y = ');

disp(solY);

% Построение графика с найденными решениями

scatter(double(solX), double(solY), 'c', 'filled');

legend('y = -0.05 \* x^2 - 2.5 \* x + 0.25', '(2 \* x^2 + 2 \* y^2 - 1)^5 - 22 \* x^2 \* y^3 = 0', ...

'vpasolve');

hold off;

% Определение функции для метода простых итераций

function [x, y, iter] = simple\_iteration\_method(eq1, x0, y0, tolerance, max\_iter)

iter = 0;

while iter < max\_iter

x = eq1(x0, y0);

y = -0.05 \* x0.^2 - 2.5 \* x0 + 0.25;

if norm([x, y] - [x0, y0]) < tolerance

return;

end

x0 = x;

y0 = y;

iter = iter + 1;

end

x = NaN;

y = NaN;

end

% Определение функции для метода Ньютона

function [x, y, iter] = newton\_method(eq1, x0, y0, tolerance, max\_iter, options)

iter = 0;

while iter < max\_iter

J = [10 \* (2 \* x0^2 + 2 \* y0^2 - 1)^4 \* 4 \* x0 \* (2 \* x0^2 + 2 \* y0^2 - 1), ...

10 \* (2 \* x0^2 + 2 \* y0^2 - 1)^4 \* 4 \* y0^2 - 66 \* x0 \* y0^3;

-0.1 \* x0 - 2.5, -1];

F = [-22 \* x0^2 \* y0^3 + (2 \* x0^2 + 2 \* y0^2 - 1)^5;

-0.05 \* x0^2 - 2.5 \* x0 + 0.25 - y0];

delta = J \ (-F);

x = x0 + delta(1);

y = y0 + delta(2);

if norm([delta(1), delta(2)]) < tolerance

return;

end

x0 = x;

y0 = y;

iter = iter + 1;

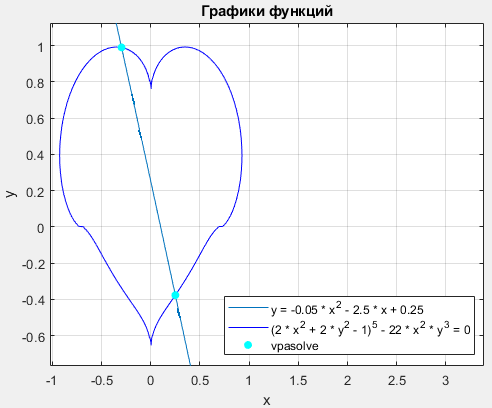
end

x = NaN;

y = NaN;

end

**Результаты расчётов**

****

Метод простых итераций не сошелся

Метод Ньютона:

x = -0.29759

y = 0.98955

Количество итераций: 88

Функция fsolve:

x = -0.29759

y = 0.98954

Метод символьной математики vpasolve:

x =

- 0.1039924174163178981251425396383 + 0.035893136564133146813877059202938i

- 0.1039924174163178981251425396383 - 0.035893136564133146813877059202938i

- 50.074473006958297489039149506854 - 19.79277400535931844001332744968i

- 50.074473006958297489039149506854 + 19.79277400535931844001332744968i

- 50.244364226729059439463082025334 - 19.901344277722736146310052073691i

- 50.244364226729059439463082025334 + 19.901344277722736146310052073691i

- 49.917956587890499266553244316159 + 19.92165909881236525405680944512i

- 49.917956587890499266553244316159 - 19.92165909881236525405680944512i

- 50.194705841206151589730679946424 - 20.096217452862564842825202914463i

- 50.194705841206151589730679946424 + 20.096217452862564842825202914463i

- 49.993299284140298023870518914916 + 20.110075786939643826981172443189i

- 49.993299284140298023870518914916 - 20.110075786939643826981172443189i

- 0.18355589573719587850201938967156 - 0.10828864777838145869042367999801i

- 0.18355589573719587850201938967156 + 0.10828864777838145869042367999801i

0.28751132049370593330333611379763 + 0.098724469832900692756029739846274i

0.28751132049370593330333611379763 - 0.098724469832900692756029739846274i

0.44897299277888320552173277909499 + 0.090403997000953188818059259860893i

0.44897299277888320552173277909499 - 0.090403997000953188818059259860893i

-0.29758884304485717091393457316552

0.24931473665531806383147006537699

y =

0.50950473825941083599953314132815 - 0.089359580006337043372843694340718i

0.50950473825941083599953314132815 + 0.089359580006337043372843694340718i

19.651235312527363615248859204468 - 49.629337753020809518512739416264i

19.651235312527363615248859204468 + 49.629337753020809518512739416264i

19.439278942434481566343455891173 - 50.239678354836291202242493350066i

19.439278942434481566343455891173 + 50.239678354836291202242493350066i

20.29839703666489841070843104783 + 49.640703658296028347539098078058i

20.29839703666489841070843104783 - 49.640703658296028347539098078058i

19.954237674391528949779724194705 - 50.631828724578547179125044024377i

19.954237674391528949779724194705 + 50.631828724578547179125044024377i

20.487506952492410064332041761581 + 50.261714276972574050489755970801i

20.487506952492410064332041761581 - 50.261714276972574050489755970801i

0.70779142256187899803244176515506 + 0.26873391747184059521284860837148i

0.70779142256187899803244176515506 - 0.26873391747184059521284860837148i

- 0.47242411315767719178122167811006 - 0.24964961485092156330226733322486i

- 0.47242411315767719178122167811006 + 0.24964961485092156330226733322486i

- 0.88210267522576195169692435709926 - 0.23006888781165208554631460424136i

- 0.88210267522576195169692435709926 + 0.23006888781165208554631460424136i

0.98954415163690409547548272040407

-0.37639473353397068940816466246581