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 № 3**

**ПО КУРСУ**

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

**«Решение СЛАУ»**

Ведущий преподаватель

Доцент кафедры ИКТ Филиппова Е.Б.

**СТУДЕНТ группы КС-20** Мелехин А.А.

**Москва**

**2024**

**Задание**

**(вариант 14):** Кажется, что Темный лорд уже почти побежден, а многие крестражи найдены и уничтожены. И теперь Гарри Поттер знает где найти следующий. Его цель диадема Кандиды Когтевран, которая находится Выручай-комнате. Но кто-то наложил на комнату заклятие вредности, и теперь она откроется только тому, кто решит систему уравнений. Вот эта система:

Решите систему и помогите спасти школу и остановить «Того-Кого-Нельзя-Называть».

Перед тем как решить задачу необходимо определить детерминант матрицы коэффициентов, её ранг, норму и число обусловленности. Задать точность решения системы уравнений. Предлагается решить систему методами простой итерации, Зейделя и Якоби, а затем сравнить результаты. Перед применением разных методов нужно проверять выполнение условий их сходимости, а в входе выполнения - фиксировать число требуемых итераций. По проделанной работе необходимо подготовить отчет с кодом и описанием методов, полученных результатов и выводами об эффективности их использования. Также решить задачу, используя оператор linsolve(A,b).

**Код программы (lab4.m)**

clear; clc;

A = [-0.76, -0.04, 0.21, -0.18; 0.45, -1.23, 0.66, 0; 0.26, 0.34, -1.11, 0; 0.05, -0.26, 0.34, -1.12]; % для проверки работоспособности методов

B = [-1.24; 0.88; -0.63; 1.17]; % для проверки работоспособности методов

% вариант 14

%A = [8 4 -6 0; 1 2 1 -6; -3 -6 -2 -9; 4 3 2 1];

%B = [596; 262.02; -731.47; 396.83];

% 1 Определение детерминанта матрицы коэффициентов

det\_A = det(A);

disp(['Детерминант матрицы коэффициентов: ', num2str(det\_A)]);

% 2 Ранг матрицы коэффициентов, норма, число обусловленности

rank\_A = rank(A);

norm\_A = norm(A);

cond\_A = cond(A);

disp(['Ранг матрицы коэффициентов: ', num2str(rank\_A)]);

disp(['Норма матрицы коэффициентов: ', num2str(norm\_A)]);

disp(['Число обусловленности матрицы коэффициентов: ', num2str(cond\_A)]);

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

% 3 Задание точности решения системы

eps = 1e-3;

disp(['Точность решения системы: ', num2str(eps)]);

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

% 4 Решить систему методом простых итераций

% k - максимальное количество итераций (иначе бесконечный цикл может быть)

k = 1000;

% Проверка условий сходимости для метода простой итерации

T\_simple = inv(diag(diag(A))) \* (diag(diag(A)) - A);

spectral\_radius\_simple = max(abs(eig(T\_simple)));

% спектр рад - максимум абсолютных значений её собственных значений

if spectral\_radius\_simple >= 1

disp('Метод простых итераций расходится');

else

disp('Метод простых итераций сходится');

t = 0.5; % множитель для метода простых итераций

[X\_simple, iters\_simple] = simple\_iteration\_method(A, B, k, eps, t);

disp('Решение методом простых итераций:');

disp(X\_simple);

disp(['Число итераций для решения: ', num2str(iters\_simple)]);

end

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

% 5. Решение системы методом Зейделя

% Проверка условий сходимости для метода Зейделя

k = 1000;

[X\_seidel, iters\_seidel] = seidel\_method(A, B, k, eps);

if ~any(~isnan(X\_seidel(:)))

disp('Метод Зейделя расходится');

else

disp('Метод Зейделя сходится');

disp('Решение методом Зейделя:');

disp(X\_seidel);

disp(['Число требуемых итераций: ', num2str(iters\_seidel)]);

end

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

% 6 Решение систему методом Якоби

% Проверка условий сходимости для метода Якоби

%if isdiagonaldominant(A)

[X\_jacobi, iters\_jacobi] = jacobi\_method(A, B, k, eps);

if ~any(~isnan(X\_jacobi(:)))

disp('Метод Якоби расходится');

else

disp('Метод Якоби сходится');

disp('Решение методом Якоби:');

disp(X\_jacobi);

disp(['Число требуемых итераций: ', num2str(iters\_jacobi)]);

end

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

disp('Решение linsolve()')

X = linsolve(A, B);

disp(transpose(X));

disp('Решение: ')

for i = 1:rank\_A

val = sprintf('x%u = %f \n', i, X(i));

fprintf(val);

end

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

function [X, iters] = seidel\_method(A, B, max\_iters, epsilon) % метод зейделя

X = zeros(size(B)); % Начальное приближение

for iters = 1:max\_iters

X\_old = X;

for i = 1:length(B)

sigma = A(i, 1:i-1) \* X(1:i-1) + A(i, i+1:end) \* X\_old(i+1:end);

X(i) = (B(i) - sigma) / A(i, i);

end

if norm(X - X\_old, inf) < epsilon % проверка условия остановки(норма разности меньше точности)

break;

end

end

end

function [X, iters] = jacobi\_method(A, B, max\_iters, epsilon) % метод якоби

X = zeros(size(B)); % Начальное приближение

n = length(B);

for iters = 1:max\_iters

X\_old = X;

for i = 1:n

% сумма всех элементов строки матрицы A,

% умноженных на соответствующие значения X\_old,

% за исключением диагонального элемента, который вычитается из суммы, домноженный на соответствующее значение X\_old.

sigma = A(i, :) \* X\_old - A(i, i) \* X\_old(i);

X(i) = (B(i) - sigma) / A(i, i);

end

if norm(X - X\_old, inf) < epsilon % проверка условия остановки(норма разности меньше точности)

break;

end

end

end

function [X, iters] = simple\_iteration\_method(A, B, max\_iters, epsilon, t) % метод простых итераций

X = zeros(size(B)); % Начальное приближение

% max\_iters - максимальное количество итераций (иначе бесконечный цикл может быть)

for iters = 1:max\_iters

X\_new = A \* X + t\*B; % последовательное нахождение нового приближения

if norm(X\_new - X, inf) < epsilon % проверка условия остановки(норма разности меньше точности)

X = X\_new;

break;

end

X = X\_new;

end

end

**Результаты работы программы**

Детерминант матрицы коэффициентов: 1302

Ранг матрицы коэффициентов: 4

Норма матрицы коэффициентов: 13.4825

Число обусловленности матрицы коэффициентов: 7.9358

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

Точность решения системы: 0.001

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

Метод простых итераций расходится

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

Метод Зейделя расходится

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

Метод Якоби расходится

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

Решение linsolve()

11.0000 121.0000 -4.0000 -2.1700

Решение:

x1 = 11.000000

x2 = 121.000000

x3 = -4.000000

x4 = -2.170000

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