|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

**Отчет**

|  |  |
| --- | --- |
| **по домашнему заданию №** | 1 |

**Дисциплина:**  Вычислительная математика

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ6-62Б |  |  | И.С. Марчук | |
|  | (Группа) |  | (Подпись, дата) | | (И.О. Фамилия) |
|  |  |  |  | |  |
| Преподаватель |  |  |  | |  |
|  |  |  | (Подпись, дата) | | (И.О. Фамилия) |

Москва, 2022

**Вариант 18.**

**Введение**

**Цель работы:** Изучение метода Гаусса численного решения квадратной СЛАУ с невырожденной матрицей, оценка числа обусловленности матрицы и исследование его влияния на погрешность приближенного решения. Изучение метода прогонки решения СЛАУ с трехдиагональной матрицей.

**Часть 1**

**Задание**

**-** реализовать метод Гаусса решения СЛАУ;

- провести решение двух заданных СЛАУ методом Гаусса, вычислить нормы невязок полученных приближенных решений, их абсолютные и относительные погрешности (использовать 1-норму и бесконечную норму);

- сравнить полученные результаты с результатами, полученными при использовании встроенной процедуры метода Гаусса;

- с использованием реализованного метода Гаусса найти A1-1 и A2-1. Проверить выполнение равентв Ai\*Ai-1 = E;

- для каждой системы оценить порядок числа обусловленности матрицы системы и сделать вывод о его влиянии на точность полученного приближенного решения и отвечающую ему невязку.

**Ход работы**

Метод Гаусса — классический метод решения системы линейных алгебраических уравнений (СЛАУ). Это метод последовательного исключения переменных, когда с помощью элементарных преобразований система уравнений приводится к равносильной системе треугольного вида (прямой ход), из которой последовательно, начиная с последних (по номеру), находятся все переменные системы (обратный ход).

Код программы в GNU Octave, содержащий исходные данные:

**% === Исходные данные ===**

A1 = [

104.0000 7.8300 -2.3100 -3.1700;

-3.7700 68.8000 8.0600 -2.7100;

2.0700 -1.7300 -103.6000 -1.1800;

-3.9600 -8.6100 9.8800 60.6000

];

F1 = [131.6200; -262.9500; -1233.9500; 335.4900];

D1 = [2; -5; 12; 3];

A2 = [

-538.0480 2696.9000 -964.6240 -80.4480;

-134.5100 674.2170 -241.1560 -20.1120;

-33.6320 168.5720 -60.2880 -5.0280;

-505.0990 2538.1660 -920.9250 -76.2090

];

F2 = [5532.8240; 1383.1680; 345.8840; 5107.4740];

D2 = [5; 6; 8; 3];

**% === Метод Гаусса ===**

function G = Gauss(A,F)

[imax, jmax] = size(A);

AF = [A F];

**% Приведение матрицы к треугольной методом Гаусса**

**% AF = rref([AF])**

for j = 1:jmax

**% Нормализация**

AF(j, :) = AF(j, :) / AF(j, j); for i = [1:j-1 j+1:imax]

AF(i, :) = AF(i, :)- (AF(j, :) \* AF(i, j));

end

end

disp('Решение СЛАУ методом Гаусса:')

X = AF(:, 5:1:end);

disp(X);

G = X;

end

function [G] = Gauss2(A)

N = size(A, 1);

E = eye(N);

G = Gauss(A, E);

end

**% === Вычисление значений ===**

disp('Решение СЛАУ 1 реализованным методом:');

X1 = Gauss(A1,F1);

disp('Решение СЛАУ 2 реализованным методом:');

X2 = Gauss(A2,F2);

disp(' ');

disp('Решение СЛАУ 1 встроенным методом:');

X11 = A1 \ F1 ;

disp(X11);

disp('Решение СЛАУ 2 встроенным методом:');

X22 = A2 \ F2 ;

disp(X22);

disp(' ');

R1 = F1 - A1\*X1;

disp('Невязка первой матрицы: ');

disp(R1);

R2 = F2 - A2\*X2;

disp('Невязка второй матрицы: ');

disp(R2);

disp(' ')

disp('Нормы невязки СЛАУ 1:');

norm1\_1 = norm(R1, 1)

norm1\_inf = norm(R1', inf)

disp(' ');

disp('Нормы невязки СЛАУ 2:');

norm2\_1 = norm(R2, 1)

norm2\_inf = norm(R2, inf)

disp(' ')

disp('Абсолютные и относительные погрешности решений СЛАУ1')

%Абсолютная погрешность по единочной норме

abs\_norm1\_1 = norm(D1-X1,1)

%Абсолютная погрешность по бесконечной норме

abs\_norm1\_inf = norm(D1-X1,inf)

%Относительная погрешность по единочной норме

delta\_norm1\_1 = norm(D1-X1,1)/norm(D1,1)

%Относительная погрешность по бесконечной норме

delta\_norm1\_inf = norm(D1-X1,inf)/norm(D1,inf)

disp(' ')

disp('Абсолютные и относительные погрешности решений СЛАУ2')

%Абсолютная погрешность по единочной норме

abs\_norm2\_1 = norm(D2-X2,1)

%Абсолютная погрешность по бесконечной норме

abs\_norm2\_inf = norm(D2-X2,inf)

%Относительная погрешность по единочной норме

delta\_norm2\_1 = norm(D2-X2,1)/norm(D2,1)

%Относительная погрешность по бесконечной норме

delta\_norm2\_inf = norm(D2-X2,inf)/norm(D2,inf)

format bank

disp(' ')

disp('A1^-1: ');

A1\_1 = Gauss2(A1);

disp('A1^-1 \* A1:');

disp(A1\*A1\_1);

disp(' ')

disp('A2^-1: ');

A2\_1 = Gauss2(A2);

disp('A2^-1 \* A2:');

disp(A2\*A2\_1);

format short

disp(' ')

disp('Числа обусловленности СЛАУ 1:');

cond1\_1 = cond(A1,1);

fprintf('Число обусловленности, с помощью единичной нормы: %6.3f \n',cond1\_1);

cond1\_inf = cond(A1,inf);

fprintf('Число обусловленности, с помощью бесконечной нормы: %6.3f \n',cond1\_inf);

disp(' ')

fprintf('Числа обусловленности СЛАУ 2: \n');

cond2\_1 = cond(A2, 1) ;

fprintf('Число обусловленности, с помощью единичной нормы: %6.3f \n',cond2\_1);

cond2\_inf = cond(A2, inf);

fprintf('Число обусловленности, с помощью бесконечной нормы: %6.3f \n',cond2\_inf);

Результаты работы программы:

1) Решения СЛАУ реализованным методом

Решение СЛАУ 1 реализованным методом:

Решение СЛАУ методом Гаусса:

2.0000

-5.0000

12.0000

3.0000

Решение СЛАУ 2 реализованным методом:

Решение СЛАУ методом Гаусса:

5.0000

6.0000

8.0000

3.0000

2) Результатами при использовании встроенного метода решения СЛАУ

Решение СЛАУ 1 встроенным методом:

2.0000

-5.0000

12.0000

3.0000

Решение СЛАУ 2 встроенным методом:

5.0000

6.0000

8.0000

3.0000

Невязка первой матрицы:

0.0000e+00

-5.6843e-14

-2.2737e-13

-5.6843e-14

Невязка второй матрицы:

9.0949e-13

2.2737e-13

1.1369e-13

-1.8190e-12

3) Нормы невязок полученных решений

Нормы невязки СЛАУ 1:

norm1\_1 = 3.4106e-13

norm1\_inf = 2.2737e-13

Нормы невязки СЛАУ 2:

norm2\_1 = 3.0695e-12

norm2\_inf = 1.8190e-12

4) Погрешности решений СЛАУ

Абсолютные и относительные погрешности решений СЛАУ1

abs\_norm1\_1 = 3.9968e-15

abs\_norm1\_inf = 1.7764e-15

delta\_norm1\_1 = 1.8167e-16

delta\_norm1\_inf = 1.4803e-16

Абсолютные и относительные погрешности решений СЛАУ2

abs\_norm2\_1 = 1.8164e-06

abs\_norm2\_inf = 8.0165e-07

delta\_norm2\_1 = 8.2565e-08

delta\_norm2\_inf = 1.0021e-07

5) Обратные матрицы

A1^-1:

Решение СЛАУ методом Гаусса:

0.01 -0.00 -0.00 0.00

0.00 0.01 0.00 0.00

0.00 -0.00 -0.01 -0.00

0.00 0.00 0.00 0.02

A1^-1 \* A1:

1.00 0.00 -0.00 -0.00

0.00 1.00 0.00 -0.00

0.00 -0.00 1.00 -0.00

-0.00 0.00 0.00 1.00

A2^-1:

Решение СЛАУ методом Гаусса:

748571.68 -2605146.40 -1553259.01 -217.87

187174.17 -651411.60 -388314.75 -54.47

46231.04 -160852.90 -96078.69 -13.62

713839.37 -2485277.03 -1477206.93 -205.52

A2^-1 \* A2:

1.00 -0.00 -0.00 -0.00

0.00 1.00 -0.00 0.00

-0.00 -0.00 1.00 0.00

-0.00 -0.00 0.00 1.00

6) Числа обусловленности

Числа обусловленности СЛАУ 1:

Число обусловленности, с помощью единичной нормы: 2.230

Число обусловленности, с помощью бесконечной нормы: 2.478

Числа обусловленности СЛАУ 2:

Число обусловленности, с помощью единичной нормы: 35875680873.638

Число обусловленности, с помощью бесконечной нормы: 21002892260.023

**Часть 2**

**Задание**

- реализовать метод прогонки;

- проверить выполнение достаточных условий применимости для системы из своего варианта;

- провести численное решение системы из своего варианта методом прогонки найти норму его невязки;

- экспериментально проверить устойчивость найденного решения к малым возмущениям исходных данных, для чего изменить несколько коэффициентов в правой части на +0,01, найти решение возмущенной системы и сравнить его с исходным.

**Ход работы**

Метод прогонки используется для решения систем линейных уравнений вида Ax=F, где A — трёхдиагональная матрица. Представляет собой вариант метода последовательного исключения неизвестных.

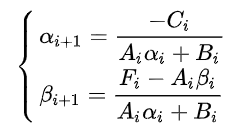
Система уравнений Ax=F равносильна соотношению:

Aixi-1 + Bixi + Cixi+1 = Fi

Метод прогонки основывается на предположении, что искомые неизвестные связаны рекуррентным соотношением:

xi = αi+1xi+1 + βi+1 , где i = n-1, n-2, ..., 1.

Коэффициенты определяются следующими выражениями:



Код программы, содержащий исходные данные:

**% === Исходные данные ===**

a = [1; 0; -1; 0; -1; 1];

b = [67; 83; 123; 75; 154; 47; 71];

c = [2; -1; 0; 1; 1; 1];

d = [13; 8; 12; 8; 18; 8; 10];

A = diag(a, -1) + diag(b, 0) + diag(c, 1);

**% === Проверка условий применимости метода ===**

function Check(a, b, c)

res = 'TRUE';

for i=1 : 6

if abs(c(i)) < abs(a(i)) + abs(b(i))

res = 'FALSE';

end

end

disp(res);

end

disp(A);

N = size(A, 1);

**% === Вычисления ===**

a = [0; a]; % Добавляем элемент в начало вектора a

c = [c; 0]; % Добавляем элемент в конец вектора с

% Прямая прогонка

alpha = zeros(N, 1); % Инициализация нулями

beta = zeros(N, 1);

y = zeros(N, 1);

y(1) = b(1);

alpha(1) = -c(1) / y(1);

beta(1) = d(1) / y(1);

for i = 2:N

y(i) = b(i) + a(i) \* alpha(i - 1);

alpha(i) = -c(i) / y(i);

beta(i) = (d(i) - a(i) \* beta(i - 1)) / y(i);

end

disp('Выполнение достаточных условий');

Check(alpha,beta,y);

% Обратная прогонка

x = zeros(N, 1);

x(N) = beta(N);

for i = 1:N-1

x(N-i) = alpha(N-i) \* x(N-i + 1) + beta(N-i);

end

% Вывод значений

disp('Альфа');

fprintf(' %f ',alpha);

fprintf('\n');

disp('Бета');

fprintf(' %f ',beta);

fprintf('\n');

disp('Х');

disp(x);

r=d-A\*x;

disp('Невязка');

disp(r);

norm\_1 = norm(r, 1) ;

disp('Единичная норма невязки:');

disp(norm\_1)

norm\_inf = norm(r, inf);

disp('Бесконечная норма невязки:');

disp(norm\_inf)

**% === Проверка устойчивости решения ===**

disp('===============================')

d1=d+0.01;

% Прямая прогонка

alpha = zeros(N, 1); % Инициализация нулями

beta = zeros(N, 1);

y = zeros(N, 1);

y(1) = b(1);

alpha(1) = -c(1) / y(1);

beta(1) = d1(1) / y(1);

for i = 2:N

y(i) = b(i) + a(i) \* alpha(i - 1);

alpha(i) = -c(i) / y(i);

beta(i) = (d(i) - a(i) \* beta(i - 1)) / y(i);

end

x = zeros(N, 1);

x(N) = beta(N);

for i = 1:N-1

x(N-i) = alpha(N-i) \* x(N-i + 1) + beta(N-i);

end

%Вывод значений

disp('Устойчивость к малым возмущениям');

disp('Альфа');

fprintf(' %f ',alpha);

fprintf('\n');

disp('Бета');

fprintf(' %f ',beta);

fprintf('\n');

disp('Х');

disp(x);

Результаты работы программы:

67 2 0 0 0 0 0

1 83 -1 0 0 0 0

0 0 123 0 0 0 0

0 0 -1 75 1 0 0

0 0 0 0 154 1 0

0 0 0 0 -1 47 1

0 0 0 0 0 1 71

1)  Выполнение достаточных условий применимости

Выполнение достаточных условий

TRUE

Альфа

-0.029851 0.012053 -0.000000 -0.013333 -0.006494 -0.021274 -0.000000

Бета

0.194030 0.094082 0.097561 0.107967 0.116883 0.172676 0.138455

2) Решение и норма его невязки

Х

0.191186

0.095258

0.097561

0.106424

0.115781

0.169730

0.138455

Невязка

1.7764e-15

0.0000e+00

0.0000e+00

0.0000e+00

0.0000e+00

1.7764e-15

0.0000e+00

Единичная норма невязки:

3.5527e-15

Бесконечная норма невязки:

1.7764e-15

===============================

Устойчивость к малым возмущениям

Альфа

-0.029851 0.012053 -0.000000 -0.013333 -0.006494 -0.021274 -0.000000

Бета

0.194179 0.094080 0.097561 0.107967 0.116883 0.172676 0.138455

3) Решение системы с малыми возмущениями

Х

0.191336

0.095256

0.097561

0.106424

0.115781

0.169730

0.138455

**Вывод:**

1) В ходе первой части работы была разработана программная реализация метода Гаусса. Решение заданных СЛАУ показало: чем больше число обусловленности матрицы, тем меньше точность ее приближенного решения и тем больше соответствующее невязки.

2) В ходе второй части работы была разработана программная реализация метода прогонки. Решение заданной СЛАУ без возмущений и с малыми возмущениями показало, что данная СЛАУ устойчива к возмущениям.