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**

О Т Ч Е Т

по домашнему заданию № 1

Дисциплина: Вычислительная математика

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

Москва, 2022

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 A_1^{-1} и A_2^{-1}. Проверить выполнение равентв $A_i*A_i^{-1} = E$;

- 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 = [
31.2000        -1.3200         -7.6800          4.0900;
7.2300       -126.0000          7.1400          3.0400;
9.4900          6.4000          6.0000          8.4500;
2.6800         -3.2900          0.2800         13.4000
];

F1 = [-83.3200;    38.9000;    -56.7000;    -504.0900];

D1 = [10; 1; 30; -40];

A2 = [
16.3820         -2.0490         -41.8290          16.3920;
307.6480        -38.4660        -840.3660         312.5280;
```

```
0.4560       -0.0570       -1.1770       0.4560;
23.2720       -2.9090       -66.3090       23.8720
];

F2 = [33.6130; 710.3420; 0.9490; 57.6730];

D2 = [2; 60; -1; 5];

% === 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:
   10.00000
    1.00000
   30.00000
  -40.00000
Решение СЛАУ 2 реализованным методом:
Решение СЛАУ методом Гаусса:
    2.0000
   60.0000
   -1.0000
    5.0000
```

2) Нормы невязок полученных решений

```
Нормы невязки для СЛАУ 1:
norm1_1 =    1.2079e-13
norm1_inf =    9.2371e-14

Нормы невязки для СЛАУ 2:
norm2_1 =  0.000000000022717
norm2_inf =  0.000000000020123
```

3) Погрешности решений СЛАУ

```
Абсолютные и относительные погрешности решений СЛАУ 1
abs_norm1_1 =    2.2649e-14
abs_norm1_inf =    1.0658e-14
delta_norm1_1 =    2.7961e-16
delta_norm1_inf =    2.6645e-16

Абсолютные и относительные погрешности решений СЛАУ 2
abs_norm2_1 =  0.000000072819
abs_norm2_inf =  0.000000064270
delta_norm2_1 =  0.0000000010709
delta_norm2_inf =  0.0000000010712
```

4) Результатами при использовании встроенного метода решения СЛАУ

```
Решение СЛАУ 1 встроенным методом:
   10.00000
    1.00000
   30.00000
  -40.00000
Решение СЛАУ 2 встроенным методом:
    2.0000
   60.0000
   -1.0000
    5.0000

Нормы невязки для СЛАУ 1:
norm1_1 =    7.8160e-14
```

```
norm1_inf =    4.9738e-14

Нормы невязки для СЛАУ 2:
norm2_1 =    1.6420e-13
norm2_inf =    1.1369e-13

Абсолютные и относительные погрешности решений СЛАУ 1
abs_norm1_1 =    2.1649e-14
abs_norm1_inf =    1.0658e-14
delta_norm1_1 =    2.6728e-16
delta_norm1_inf =    2.6645e-16

Абсолютные и относительные погрешности решений СЛАУ 2
abs_norm2_1 =  0.00000015842
abs_norm2_inf =  0.00000013982
delta_norm2_1 =  0.0000000023297
delta_norm2_inf =  0.0000000023304
```

5) Обратные матрицы

```
A1^-1:
Решение СЛАУ методом Гаусса:
   0.02   0.00   0.03  -0.03
  -0.00  -0.01   0.01  -0.00
  -0.03   0.01   0.12  -0.07
  -0.00  -0.00  -0.01   0.08
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:
   -1096900.00     137100.00    2885099.99   -1096800.00
  -10201573.31    1275096.66   26827459.95  -10200759.98
     -22800.00       2850.00      59900.00     -22800.00
    -237146.67      29643.33     622945.00    -237145.00
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: 24.406
Число обусловленности, с помощью бесконечной нормы: 32.429

Числа обусловленности СЛАУ 2:
Число обусловленности, с помощью единичной нормы:
28865938592.905
Число обусловленности, с помощью бесконечной нормы:
72709218091.407
```

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:

$A_i x_{i-1} + B_i x_i + C_i x_{i+1} = F_i$

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

$x_i = \alpha_{i+1} x_{i+1} + \beta_{i+1}$, где i = n-1, n-2, ..., 1.

Коэффициенты определяются следующими выражениями:

$$\begin{cases} \alpha_{i+1} = \dfrac{-C_i}{A_i\alpha_i + B_i} \\ \beta_{i+1} = \dfrac{F_i - A_i\beta_i}{A_i\alpha_i + B_i} \end{cases}$$

Код программы, содержащий исходные данные:

```
% === Исходные данные ===

a = [-1;  1;  1; -1;  1];
b = [50; 90; 125; 110; 85; 70];
c = [1; 1; -1; 0; 1];
d = [10; -9; 12; 11; 9; 8];

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 c

% 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('X');
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('X');
disp(x);
```

Результаты работы программы:

1) Выполнение достаточных условий применимости

```
Выполнение достаточных условий
TRUE
```

2) Решение и норма его невязки

```
X
   0.201977
  -0.098840
   0.097584
   0.099113
   0.105722
   0.112775
Невязка
   0.0000e+00
   0.0000e+00
  -1.7764e-15
   0.0000e+00
   1.7764e-15
   0.0000e+00
Единичная норма невязки:
   3.5527e-15
Бесконечная норма невязки:
   1.7764e-15
```

3) Решение системы с малыми возмущениями

```
X
   0.202177
  -0.098838
   0.097584
   0.099113
   0.105722
   0.112775
```

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.