

Contents

- [Задание данных](#)
- [Прямой ход](#)
- [Обратный ход](#)
- [Вектор невязки и его первая и инфинити норма](#)
- [Расчёт абсолютной и относительной погрешностей полученного решения](#)
- [Оценка числа обусловленности](#)
- [Функции для расчёта норм](#)

Задание данных

```
clc
clear
format long                                % задаём формат на вывод
A = readmatrix('basic_matrix.txt')         % считываем матрицу из файла
A0 = A;                                    % запоминаем исходную матрицу
[n,m] = size(A);                           % узнаём размер матрицы для реализации алгоритма
b_real = A(:,m);                            % матрица правой части
```

A =

1.0e+02 *

Columns 1 through 3

0.3120000000000000	-0.0132000000000000	-0.0768000000000000
0.0723000000000000	-1.2600000000000000	0.0714000000000000
0.0949000000000000	0.0640000000000000	0.0600000000000000
0.0268000000000000	-0.0329000000000000	0.0028000000000000

Columns 4 through 5

0.0409000000000000	-0.8332000000000000
0.0304000000000000	0.3890000000000000
0.0845000000000000	-0.5670000000000000
0.1340000000000000	-5.0409000000000000

Прямой ход

```
for i = 1:n                                % цикл по всем сточкам
    [y,z] = max(abs(A(i:n,i)));             % находим максимальный элемент в столбце (по модулю)
    z = z + i-1;                             % y - макс значение z - номер строки
    vectorA = A(i,:);                         % запоминаем рабочую строку
    A(i,:) = A(z,:);                         % осуществляем перестановку строчек
    A(z,:) = vectorA;
    A;                                        % выводим матрицу на экран
    v = A(i,i);                              % выбираем диагональный элемент
    for k = i:m                               % производим деление строки на диагональный элемент
        A(i,k) = A(i,k) / v;
    end
    for k = (i+1):n                           % производим в каждой последующей строке вычитание
        l = A(k,i);                           % на соответствующий элемент нижнего столбца
        for j = i:m                           % последовательно для получения 0
            A(k,j) = A(k,j) - l * A(i,j);
        end
    end
    A;
```

```

end
end
A;

```

Обратный ход

```

for i = n:-1:1 % от нижней строчке идём к верхней
    for k = (i-1):-1:1 % от предпоследнего столбца
        l = A(k,i); % домножаем элемент нижней строчки на
        for j = m:-1:i % элемент вышестоящей строчки в одном столбце
            A(k,j) = A(k,j) - l * A(i,j); % производим вычитание по всем столбца
        end % до появления единицы
        A; % отображаем матрицу
    end
end
end

```

Вектор невязки и его первая и инфинити норма

```

x = A(:,m); % полученные решения исходной системы
x_real = readmatrix('X_real.txt'); % считываем матрицу из файла
b = A0(:,1:m-1) * x(:); % вектор правой части для полученного решения
r = b_real - b; % вычисляем вектор невязки
r_norm_1 = FirstNorm(r); % первая норма для вектора невязки
r_norm_inf = InfNorm(r); % инфинити норма для вектора невязки

sigma_b_1 = r_norm_1/FirstNorm(b_real); % относительная погрешность для первой нормы
sigma_b_inf = r_norm_inf/InfNorm(b_real); % относительная погрешность для инфинити нормы

```

Расчёт абсолютной и относительной погрешностей полученного решения

```

delta_x_1 = FirstNorm(x_real - x); % абсолютная погрешность для первой нормы
delta_x_inf = InfNorm(x_real - x); % абсолютная погрешность для инфинити нормы

sigma_x_1 = delta_x_1/FirstNorm(x); % относительная погрешность для первой нормы
sigma_x_inf = delta_x_inf/InfNorm(x); % относительная погрешность для инфинити нормы

```

Оценка числа обусловленности

```

Nu_1 = sigma_x_1/sigma_b_1; % нижняя оценка числа обусловленности по первой норме
Nu_inf = sigma_x_inf/sigma_b_inf; % нижняя оценка числа обусловленности по инфинити норме

k = cond(A0(:,1:n)); % число обусловленности (расчёт MatLab)
g = norm(inv(A0(:,1:n)))*norm(A0(:,1:n)); % число обусловленности (прямой расчёт)

```

Функции для расчёта норм

```

function norm = FirstNorm(x) %функция расчёта первой нормы
    sum = 0;
    [n,m] = size(x);
    for i = 1:n
        sum = sum + abs(x(i));
    end
    norm = sum;
end

function norm = InfNorm(x) %функция расчёта инфинити нормы
    count = 0;
    [n,m] = size(x);

```

```
for i = 1:n
    if count < abs(x(i))
        count = abs(x(i));
    end
end
norm = count;
end
```
