ЛР 6. Решение систем линейных уравнений.

③ Задайте матрицу A и вектор-столбец f системы линейных уравнений $AX = f$, используя генератор случайных чисел. Очевидно, можно получить решение таким образом: $X = A^{-1}f$ (предварительно проверив, что матрица A не вырожденная) или по правилу Крамера ($x_i = \frac{\det A_i}{\det A}$, где A_i — матрица, получающаяся из матрицы A заменой i-го столбца на столбец правой части f). Реализуйте и проверьте работоспособность этих методов. Несмотря на простоту использования в MATLAB, эти варианты чрезвычайно неэкономичны по числу операций.

Зададим матрицу и вектор-столбец:

```
A = [5, 2, 3, 2; 7, 2, -9, 0; 6, -3, 1, -2; 3, 8, -1, 2];
f = [1; 3; 2; -1];
```

```
A =

     5     2     3     2
     7     2    -9     0
     6    -3     1    -2
     3     8    -1     2


f =

     1
     3
     2
    -1
```

Найдем решение системы методом обратной матрицы:

```
X=inv(A)*f
```

```
X =

    0.3043
   -0.3478
   -0.1739
    0.3478
```

Найдем решение системы правилом Крамера:

```
A1=A;
A2=A;
A3=A;
A4=A;
A1(:,1)=f;
A2(:,2)=f;
A3(:,3)=f;
A4(:,4)=f;
x=det(A1)/det(A);
y=det(A2)/det(A);
z=det(A3)/det(A);
t=det(A4)/det(A);
X = [x; y; z; t]
```

```
X =

    0.3043
   -0.3478
   -0.1739
    0.3478
```

③ Напишите программу нахождения решения системы линейных уравнений методом Гаусса с выбором главного элемента.

Функция для решения системы линейных уравнений методом Гаусса с выбором главного элемента:

```
function x = gauss(A,b)
[n,m] = size(A);

for k = 1:n-1
    [~,p] = max(abs(A(k:n,k)));
    p = p + k - 1;
    if p ~= k
        A([k,p],:) = A([p,k],:);
        b([k,p]) = b([p,k]);
    end

    for i = k+1:n
        factor = A(i,k) / A(k,k);
        A(i,k+1:n) = A(i,k+1:n) - factor * A(k,k+1:n);
        b(i) = b(i) - factor * b(k);
    end
end

x = zeros(n,1);
x(n) = b(n) / A(n,n);
for i = n-1:-1:1
    x(i) = (b(i) - A(i,i+1:n)*x(i+1:n)) / A(i,i);
end
end
```

Решим данную систему методом Гаусса с выбором элемента:

```
ans =

    0.3043
   -0.3478
   -0.1739
    0.3478
```

③ Функция **rref** MATLAB также приводит матрицу **[A f]** к диагональному виду, из которого сразу же видно решение системы. Также пакет содержит операцию левого матричного деления, с помощью которой очень просто найти решение: `X = A\f`. Более того, эта операция позволяет решать недоопределённые и переопределённые системы линейных уравнений, выбирая алгоритм решения в зависимости от вида матрицы A.

Решим систему через функцию rref:

```
rref([A f])
```

```
ans =

    1.0000         0         0         0    0.3043
         0    1.0000         0         0   -0.3478
         0         0    1.0000         0   -0.1739
         0         0         0    1.0000    0.3478
```

Решим систему, используя операцию левого матричного деления:

```
X=A\f
```

```
X =

    0.3043
   -0.3478
   -0.1739
    0.3478
```

③ Задайте случайным образом матрицу A размерности 20×20 и вектор X. Определите число обусловленности матрицы A с помощью функции `cond`. Изменяя значения некоторых элементов матрицы A, добейтесь, чтобы её число обусловленности стало больше 10^3. Используя A и X, найдите вектор $f = AX$. Полагая вектор X неизвестным, решите систему линейных уравнений всеми предложенными выше методами и сравните найденные решения с уже известным. Какой из методов дал более точный результат? Обратите внимание на решения, полученные обычным методом Гаусса и методом с выбором главного элемента.

Заданный вектор X и зайдем число обусловленности матрицы А:

```
X =

    2.5367
    2.2936
   -1.3187
   -0.4741
    0.9874
   -2.0730
    1.2440
   -0.3152
    1.3993
   -0.8799
   -0.9225
    2.3144
    0.5880
   -0.3098
   -1.0330
    1.1034
    0.3421
   -2.2640
   -0.9033
    2.9700
```

```
ans =

  2.8954e+003
```

Решая заданную систему методом обратных матриц, получаем:

```
X =

    2.5367
    2.2936
   -1.3187
   -0.4741
    0.9874
   -2.0730
    1.2440
   -0.3152
    1.3993
   -0.8799
   -0.9225
    2.3144
    0.5880
   -0.3098
   -1.0330
    1.1034
    0.3421
   -2.2640
   -0.9033
    2.9700
```

```
Md1 =

   1.3012e-013
```

Md – погрешность метода

Используя правило Крамера:

```
x =

    2.5367
    2.2936
   -1.3187
   -0.4741
    0.9874
   -2.0730
    1.2440
   -0.3152
    1.3993
   -0.8799
   -0.9225
    2.3144
    0.5880
   -0.3098
   -1.0330
    1.1034
    0.3421
   -2.2640
   -0.9033
    2.9700
```

```
Md =

   1.5721e-013
```

Функцию rref, приведение к диагональному виду:

```
        0          0    2.5367
        0          0    2.2936
        0          0   -1.3187
        0          0   -0.4741
        0          0    0.9874
        0          0   -2.0730
        0          0    1.2440
        0          0   -0.3152
        0          0    1.3993
        0          0   -0.8799
        0          0   -0.9225
        0          0    2.3144
        0          0    0.5880
        0          0   -0.3098
        0          0   -1.0330
        0          0    1.1034
        0          0    0.3421
        0          0   -2.2640
    .0000          0   -0.9033
        0     1.0000    2.9700
```

```
Md2 =

  6.1728e-014
```

Левое матричное деление:

```
X =

    2.5367
    2.2936
   -1.3187
   -0.4741
    0.9874
   -2.0730
    1.2440
   -0.3152
    1.3993
   -0.8799
   -0.9225
    2.3144
    0.5880
   -0.3098
   -1.0330
    1.1034
    0.3421
   -2.2640
   -0.9033
    2.9700
```

```
Md3 =

  6.6613e-015
```

Метод Гаусса:

```
ans =

    2.5367
    2.2936
   -1.3187
   -0.4741
    0.9874
   -2.0730
    1.2440
   -0.3152
    1.3993
   -0.8799
   -0.9225
    2.3144
    0.5880
   -0.3098
   -1.0330
    1.1034
    0.3421
   -2.2640
   -0.9033
    2.9700
```

```
Md4 =

  4.5075e-014
```

Время потраченное на 100000 повторений для различных методов:

```
time =

    2.6052
```

Для обратных матриц

```
time =

   30.7946
```

Для правила Крамера

```
time =

  132.4136
```

Для метода Гаусса

```
time =

    2.1372
```

Для левого матричного деления

```
time =

   15.2725
```

Для rref()