OR 第一周上机作业

20123101 李昀哲

2022.12.2

1、给定一个矩阵，判断是否存在单位矩阵，并给出所在列

准备部分：

```
% Operational Research
% @author 李昀哲 20123101
% Dec 1, 2022
% Generate a mxn matrix where an eye is randomly assigned in it.
m = 3;
n = 8;
A = 10 * randn(m, n);
I = eye(m, m);
randIndex = randperm(size(A, 2));
A(:,randIndex(1:m))=I;
disp(A);
```

判断部分：

```
% Our expectation is to extract those columns which is an element of I.
for i  = 1:size(A, 2)
    for j = 1:size(A,2)
            if isequal(i, j)
                continue;
            end
        for k = 1:size(A,2)
            if or (isequal(i, k), isequal(j, k))
                continue;
            end
            temp = A(:, [i j k]);

            % Judge if temp is equal I to determine whether exist eye.
            if (isequal(temp, I))
                 break;
            end
        end
        if (isequal(temp, I))
            break;
        end
    end
        if (isequal(temp, I))
            break;
        end
end
```

结果输出：

```
% Result Output
if (isequal(temp, I))
    disp("Has eye");
    fprintf("i = %d, j = %d, k = %d\n", i, j, k);
else
    disp("No eye!")
end
```

运行结果：Has eye 即随机生成的矩阵存在单位阵，且所在列为 3，5，1

```
         0   14.3838    1.0000   -2.4145         0    6.2771   -8.6365  -11.1350
         0    3.2519         0    3.1921    1.0000   10.9327    0.7736   -0.0685
    1.0000   -7.5493         0    3.1286         0   11.0927  -12.1412   15.3263

Has eye
i = 3, j = 5, k = 1
```

2、高斯消元法求解:

A = [1 2 3; -1 3 7; 9 0 3];
b = [1 4 7]’

主函数部分：

矩阵准备：

```
% Operational Research
% @author 李昀哲 20123101
% Dec 1, 2022
A = [1 2 3; -1 3 7; 9 0 3];
b = [1 4 7]';
```

方程求解：

```
x = gauss_elim(A, b);
```

答案输出：

```
% 输出答案
for ans_num = 1:size(A,1)
    fprintf("\tx%d = %d\n", ans_num, x(size(A,1)-ans_num+1));
end
```

Gauss_elim 函数部分：

处理输入数据：

```
function[x] = gauss_elim(A,b)
    % Merge two matrix as M.
    M = [A b];
    % 明确行数和列数
    shape_M = size(M);
    disp(shape_M);
    disp (M);
```

第一版根据手动计算方式编写部分：

```
%   第一版由手动计算的过程实现，后改为上述按行、列用循环处理
%     % 第一列
%     cof = M(2,1) / M(1,1);
%     for i = 1:size(M, 2)
%         M(2,i) = M(2,i)-cof*M(1,i);
%     end
%     cof = M(3,1) / M(1,1);
%     for i = 1:size(M, 2)
%         M(3,i) = M(3,i)-cof*M(1,i);
%     end
%     disp(M);
%     % 第二列
%     cof = M(3,2) / M(1,2);
%     for i = 2:size(M, 2)
%         M(3,i) = M(3,i)-cof*M(1,i);
%     end
%     disp(M);

%     x3 = M(3,4)/M(3,3);
%     disp(M(3,4)/M(3,3));
%     x2 = (5 - 10 * M(3,4)/M(3,3)) / 5;
%     disp((5 - 10 * M(3,4)/M(3,3)) / 5);
%     disp(1 - 3*x3 - 2*x2);
```

第二版使用循环对行列进行处理：

```
% 按列进行线性变换化为行阶梯
for column = 1:shape_M(1)-1
    % 对当前列的行进行处理
    for row = column+1:shape_M(1)
        porption = M(row, column) / M(1, column);
        for column_this_row = column:shape_M(2)
            M(row, column_this_row) = M(row, column_this_row) - porption*M(1, column_this_row);
        end
    end
    disp(M);
end
```

```
% 求解
x = [];
iter = 0;
for row = shape_M(1):-1:1
    if iter ~= 0
        i = 1;
        for column = shape_M(2)-1:-1:shape_M(2)-iter
                M(row, shape_M(2)) = M(row, shape_M(2)) - M(row, column)*x(i);
                i = i + 1;
        end
    end
    iter = iter + 1;
    x = [x;M(row, shape_M(2)) / M(row, shape_M(2)-iter)];
end
```

输出结果：首先输出变换结果，再输出求解结果

```
 1     2     3     1
-1     3     7     4
 9     0     3     7

 1     2     3     1
 0     5    10     5
 0   -18   -24    -2

 1     2     3     1
 0     5    10     5
 0     0     3     7

x1 = 1.333333e+00
x2 = -3.666667e+00
x3 = 2.333333e+00
```