作业2 矩阵代数

1.设a=(1, 2, 3), b=(2, 4, 3), 分别计算a./b, a. \b, a/b, a \b, 分析结果的意义.

a./b: a中的每一个元素除以b中的对应元素

a.\b: b中的每一个元素除以a中的对应位置的元素

a/b: a左除b，即矩阵方程ax=b的解

a\b: a右除b，即矩阵方程Xa=b的解

输出：

ans = 1×3

0.5000 0.5000 1.0000

ans = 1×3

2 2 1

ans = 0.6552

ans = 3×3

0 0 0

0 0 0

0.6667 1.3333 1.0000

2.用矩阵除法解下列线性方程组，判断解的意义，并用矩阵乘法验算:

(1)

定义函数：

function solution\_status = check\_unique\_solution(A, b)

% 计算系数矩阵A的秩

rankA = rank(A);

% 构造增广矩阵[A, b]并计算其秩

rankAB = rank([A, b]);

% 判断方程组的解的情况

if rankA == rankAB

% 如果系数矩阵A和增广矩阵[A, b]的秩相等，方程组有唯一解

if rankA < size(A, 1) % 未知数个数

solution\_status = '无穷多解';

else

solution\_status = '唯一解';

end

elseif rankA < rankAB

% 如果系数矩阵A的秩小于增广矩阵[A, b]的秩，方程组无解

solution\_status = '无解';

end

end

代码：

A=[4 1 -1;3 2 -6;1 -5 3];

b=[9;-2;1];

solution\_status = check\_unique\_solution(A, b)

x=A\b

A\*x

输出：

solution\_status = '唯一解'

x = 3×1

2.3830

1.4894

2.0213

ans = 3×1

9.0000

-2.0000

1.0000

(2) ;

代码：

A=[4 -3 3;3 2 -6;1 -5 3];

b=[-1;-2;1];

solution\_status = check\_unique\_solution(A, b)

x=A\b

A\*x

输出：

solution\_status = '唯一解'

x = 3×1

-0.4706

-0.2941

0

ans = 3×1

-1

-2

1

(3) ;

代码：

A=[4 1;3 2;1 -5];

b=[1;1;1];

solution\_status = check\_unique\_solution(A, b)

x=A\b

A\*x

输出：

solution\_status = '无解'

x = 2×1

0.3311

-0.1219

ans = 3×1

1.2026

0.7497

0.9404

(4) ;

代码：

A=[2 1 -2 1;1 2 1 -1;1 1 2 1];

b=[1;2;3];

solution\_status = check\_unique\_solution(A, b)

x=A\b

A\*x

输出：

solution\_status = '唯一解'

x = 4×1

1.2727

0

0.8182

0.0909

ans = 3×1

1.0000

2.0000

3.0000

3. 求下列矩阵的行列式、逆、特征值和特征向量；

(1) (2) (3)

(1)代码：

function matrixProperties(A)

% 检查输入是否为方阵

[m, n] = size(A);

if m ~= n

error('输入必须为方阵');

end

% 计算行列式

detA = det(A);

fprintf('行列式为: %f\n', detA);

% 如果行列式不为零，计算并打印逆矩阵

if detA ~= 0

invA = inv(A);

fprintf('逆矩阵为:\n');

disp(invA);

else

fprintf('矩阵不可逆\n');

end

% 计算并打印特征值和特征向量

[V, D] = eig(A);

fprintf('特征值为:\n');

disp(diag(D));

fprintf('特征向量为:\n');

disp(V);

end

A=[4 1 -1; 3 2 -6; 1 -5 3];

matrixProperties(A)

输出：

行列式为: -94.000000

逆矩阵为:

0.2553 -0.0213 0.0426

0.1596 -0.1383 -0.2234

0.1809 -0.2234 -0.0532

特征值为:

-3.0527

3.6760

8.3766

特征向量为:

0.0185 -0.9009 -0.3066

-0.7693 -0.1240 -0.7248

-0.6386 -0.4158 0.6170

(2)代码：

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

matrixProperties(A)

输出：

行列式为: 1.000000

逆矩阵为:

2.0000 -2.0000 1.0000

1.0000 -1.0000 1.0000

2.0000 -3.0000 2.0000

特征值为:

1.0000 + 0.0000i

1.0000 + 0.0000i

1.0000 - 0.0000i

特征向量为:

0.5774 + 0.0000i 0.5773 + 0.0000i 0.5773 - 0.0000i

0.5774 + 0.0000i 0.5773 + 0.0000i 0.5773 - 0.0000i

0.5773 + 0.0000i 0.5774 + 0.0000i 0.5774 + 0.0000i

(3)代码：

A=[5 7 6 5; 7 10 8 7;6 8 10 9;5 7 9 10];

matrixProperties(A)

输出：

行列式为: 1.000000

逆矩阵为:

68.0000 -41.0000 -17.0000 10.0000

-41.0000 25.0000 10.0000 -6.0000

-17.0000 10.0000 5.0000 -3.0000

10.0000 -6.0000 -3.0000 2.0000

特征值为:

0.0102

0.8431

3.8581

30.2887

特征向量为:

0.8304 0.0933 0.3963 0.3803

-0.5016 -0.3017 0.6149 0.5286

-0.2086 0.7603 -0.2716 0.5520

0.1237 -0.5676 -0.6254 0.5209

4. 判断第3题各小题的矩阵是否可以相似对角化，如果可以，求出对角矩阵和对应的相似变换矩阵.

定义函数：

function diagonalizeMatrix(A)

% 检查输入是否为方阵

[m, n] = size(A);

if m ~= n

error('输入必须为方阵');

end

% 计算特征值和特征向量

[V, D] = eig(A);

array = eig(A)';

uniqueArray = unique(abs(array));

% 检查特征向量的个数是否等于矩阵的阶数

if rank(V) == min(m,n) && length(array) == length(uniqueArray)

fprintf('矩阵可以对角化\n');

fprintf('对角矩阵为:\n');

disp(D);

fprintf('相似变换矩阵为:\n');

disp(V);

else

fprintf('矩阵不可对角化\n有重根');

disp(array);

end

end

(1)代码：

A=[4 1 -1; 3 2 -6; 1 -5 3];

diagonalizeMatrix(A)

输出：

矩阵可以对角化

对角矩阵为:

-3.0527 0 0

0 3.6760 0

0 0 8.3766

相似变换矩阵为:

0.0185 -0.9009 -0.3066

-0.7693 -0.1240 -0.7248

-0.6386 -0.4158 0.6170

(2)代码：

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

diagonalizeMatrix(A)

输出：

矩阵不可对角化

有重根 1.0000 + 0.0000i 1.0000 - 0.0000i 1.0000 + 0.0000i

(3)代码：

A=[4 1 -1; 3 2 -6; 1 -5 3];

diagonalizeMatrix(A)

输出：

矩阵可以对角化

对角矩阵为:

0.0102 0 0 0

0 0.8431 0 0

0 0 3.8581 0

0 0 0 30.2887

相似变换矩阵为:

0.8304 0.0933 0.3963 0.3803

-0.5016 -0.3017 0.6149 0.5286

-0.2086 0.7603 -0.2716 0.5520

0.1237 -0.5676 -0.6254 0.5209

5. 求下列向量组的秩和它的一个最大线性无关组，并将其余向量用该最大无关组线性表示；

=(4, -3, 1, 3), =(2, -1, 3, 5), =(1, -1, -1, -1), =(3, -2, 3, 4), =(7, -6, -7, 0).

代码：

function vectorProperties(V)

% 求向量组的秩

r = rank(V);

fprintf('向量组的秩为: %d\n', r);

% R: 行化简后的阶梯型; j: 主元

[R, j] = rref(V);

% 打印最大线性无关组

fprintf('最大线性无关组为:\n');

disp(V(:,j));

num\_vectors = size(V, 2);

for a = setdiff(1:num\_vectors, j)

% 输出当前向量的编号

fprintf("a%d用该最大无关组线性表示为\n", a);

term\_count = 1;

% 遍历当前向量对应的系数

for b = R(:, a)'

if b ~= 0

% 格式化输出系数和向量编号

if b > 0 && term\_count ~= 1

fprintf("+%d\*a%d", round(b), term\_count)

else

fprintf("%d\*a%d", round(b), term\_count)

end

% 更新向量编号

term\_count = term\_count + 1;

end

end

fprintf("\n");

end

end

a1 = [4;-3; 1; 3];

a2 = [2; -1; 3; 5];

a3 = [1; -1; -1; -1];

a4 = [3; -2; 3; 4];

a5 = [7; -6; -7; 0];

A = [a1,a2,a3,a4,a5];

vectorProperties(A);

输出：

向量组的秩为: 3

最大线性无关组为:

4 2 3

-3 -1 -2

1 3 3

3 5 4

a3用该最大无关组线性表示为

1\*a1-1\*a2

a5用该最大无关组线性表示为

5\*a1+1\*a2-5\*a3

6. (二次型标准化) 用正交变换化下列二次型为标准形:

.

代码：

A = [1 -2 2;-2 -2 4;2 4 -2]

[Q,D]=schur(A)

输出：

A = 3×3

1 -2 2

-2 -2 4

2 4 -2

Q = 3×3

0.3333 0.8944 -0.2981

0.6667 -0.4472 -0.5963

-0.6667 0 -0.7454

D = 3×3

-7.0000 0 0

0 2.0000 0

0 0 2.0000

经过正交变换X=QY，二次型化为标准型：