VLSI\_DSP HW1

4108064101 杜冠廷

**Q1**

Matlab code:

|  |
| --- |
| clear all %clear workspace  clc %clear command window  %clf %clear figure  A = [ 15 -13 20 -8;  -5 -15 -4 -4;  -17 16 -2 9;  10 -19 -14 -15;  -7 8 -7 15;  14 10 -8 -17;  -5 -3 16 -2;  13 -5 -10 -19];  b = [13; 10; -15; 9; 3; 18; 3; 20];  A\_plus = pinv(A);  x\_a = A\_plus \* b;  [Q, R] = qr(A);  R\_u = R([1 2 3 4], : );  y = Q' \* b; % Q' = Q transpose  y\_u = y([1 2 3 4], : );  x\_b = inv(R\_u) \* y\_u; % x\_b = R \ y = R \ (Q \ b) |

Ans:

|  |  |
| --- | --- |
| 1. Pseudo inverse | 1. QR decomposition |
| 1. The results of a) and b) are same. | |

**Q2**

Matlab code:

|  |
| --- |
| clear all %clear workspace  clc %clear command window  M = [ -2 16 -6 -16 3 15 -6 -19;  16 -17 10 -2 7 8 3 5;  -6 10 15 -1 -15 -18 9 -8;  -16 -2 -1 9 0 0 0 18;  3 7 -15 0 14 19 -12 11;  15 8 -18 0 19 10 -8 -17;  -6 3 9 0 -12 -8 15 20;  -19 5 -8 18 11 -17 20 20];    [V, D]=eig(M);    for i = 1:2000  [Q\_mat, R\_mat] = QR\_decomposition\_func(M);  M = R\_mat \* Q\_mat;  end |
| function [Q, R] = QR\_decomposition\_func(M)  s = 2; % start row, adding by 1 at each round beginning  R = M;  Q = eye(8, 8);% eye(8, 8) = identity matrix 8\*8  % QR decomposition  for j = 1:8  for i = s:8  rotation\_param = R(:, j);  cos = rotation\_param(j) / (rotation\_param(j)^2 + rotation\_param(i)^2)^0.5;  sin = rotation\_param(i) / (rotation\_param(j)^2 + rotation\_param(i)^2)^0.5;  % creat unitary matrix q  q = eye(8, 8);  q(i, i) = cos;  q(j, i) = sin;  q(i, j) = -sin;  q(j, j) = cos;  % cal & update R  R = q \* R;  Q = q \* Q;  end  s = s + 1;  end  Q = Q';  end |

Ans:

|  |
| --- |
| Eigen value decomposition using a QR decomposition based iterative algorithm |
| Using the function [V,D] = eig(M) |

------------------------------------------------------我是分隔線------------------------------------------------------

*Refence*

Q2

Panju, Maysum. "Iterative methods for computing eigenvalues and eigenvectors." *arXiv preprint arXiv:1105.1185* (2011).