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:

a) Pseudo inverse

	1
1	0.4638
2	-0.1005
3	-0.0716
4	-0.4137

b) QR decomposition

	1
1	0.4638
2	-0.1005
3	-0.0716
4	-0.4137

c) 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

	1	2	3	4	5	6	7	8
1	67.1862	9.2886e-15	-4.9782e-15	6.6778e-15	5.0235e-15	5.8378e-15	-7.1189e-15	-8.0300e-15
2	1.1924e-313	46.8129	7.2839e-15	-8.1839e-15	-3.2820e-15	-1.2214e-15	2.4889e-15	2.9071e-16
3	-5.1000e-3...	-1.0442e-2...	-36.9137	-2.5404e-15	3.9674e-15	-1.1017e-14	1.6999e-15	1.4317e-16
4	7.9000e-323	-1.4300e-3...	-1.9330e-3...	-26.0043	9.7076e-15	1.2770e-15	-1.4281e-15	-3.4904e-15
5	-1.0000e-3...	7.4000e-323	-1.5000e-3...	3.5000e-323	16.4684	5.3828e-15	-3.9886e-15	1.0991e-15
6	-1.6300e-3...	1.2000e-322	8.9000e-323	0	1.3300e-322	-11.0351	1.3230e-15	-1.9814e-15
7	4.9000e-324	-8.4000e-3...	2.0000e-323	-3.0000e-3...	2.5000e-323	-2.0000e-3...	6.0581	1.1911e-15
8	-7.9000e-3...	-5.4000e-3...	0	2.0000e-323	4.0000e-323	-2.0000e-3...	4.9000e-324	1.4275

Using the function [V,D] = eig(M)

	1	2	3	4	5	6	7	8
1	-36.9137	0	0	0	0	0	0	0
2	0	-26.0043	0	0	0	0	0	0
3	0	0	-11.0351	0	0	0	0	0
4	0	0	0	1.4275	0	0	0	0
5	0	0	0	0	6.0581	0	0	0
6	0	0	0	0	0	16.4684	0	0
7	0	0	0	0	0	0	46.8129	0
8	0	0	0	0	0	0	0	67.1862

--我是分隔線--

Refence

Q2

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