
Name: Ishaan Malhotra

Table of Contents

Roll no. : 1610110152	1
Instructor: Prof. Vijay Chakka	1
Lab 5	1
Aim: Spectral methods for graph clustering	1
Question 1	1
Question 2	5
Question 3	5

Roll no. : 1610110152

Instructor: Prof. Vijay Chakka

Lab 5

Aim: Spectral methods for graph clustering

```
clc
clear all
close all
```

Question 1

```
A = [0 0.23 0.74 0.24 0 0 0 0; 0.23 0 0.35 0 0.23 0 0 0; 0.74 0.35
0 0.26 0.24 0 0 0; 0.24 0 0.26 0 0 0 0.32 0; 0 0.23 0.24 0 0 0.51 0
0.14; 0 0 0 0 0.51 0 0 0.15; 0 0 0 0.32 0 0 0 0.32; 0 0 0 0 0.14 0.15
0.32 0];
lineA = LineGraph(60);
[A2,xy] = bucky;

Laplacian_A = laplacianMat(A);
Laplacian_lineA = laplacianMat(lineA);
Laplacian_bucky = laplacianMat(A2);

[U1,Lam1] = eig(Laplacian_A);
[U2,Lam2] = eig(Laplacian_lineA);
[U3,Lam3] = eig(Laplacian_bucky);

[sorted1, i1] = sort(diag(Lam1));
U1_0 = U1(:,i1(1));
U1_1 = U1(:,i1(2));
```

```
U1_7 = U1(:,i1(8));

[sorted2, i2] = sort(diag(Lam2));
U2_0 = U2(:,i2(1));
U2_1 = U2(:,i2(2));
U2_59 = U2(:,i2(60));

[sorted3, i3] = sort(diag(Lam3));
U3_0 = U3(:,i3(1));
U3_1 = U3(:,i3(2));
U3_59 = U3(:,i3(60));

% Reference graph plots
randA = rand(length(A),3);
figure;

subplot(2,2,1);
plot3DGraph(A,randA)
title('Graph from ref. paper')

subplot(2,2,2);
plot3Dv2(A,randA,U1_0);
title('Eigenvector U(0)')

subplot(2,2,3);
plot3Dv2(A,randA,U1_1);
title('Eigenvector U(1)')

subplot(2,2,4);
plot3Dv2(A,randA,U1_7);
title('Eigenvector U(7)')

% Line plots
figure;

subplot(3,1,1);
plotLine3D(lineA,U2_0);
title('Eigenvector U(0)')

subplot(3,1,2);
plotLine3D(lineA,U2_1);
title('Eigenvector U(1)')

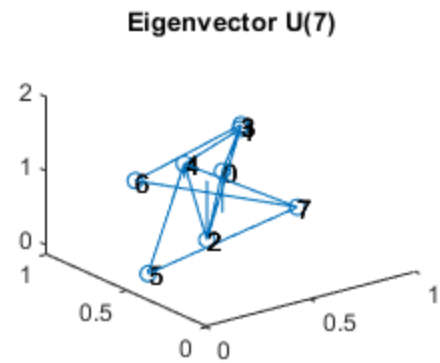
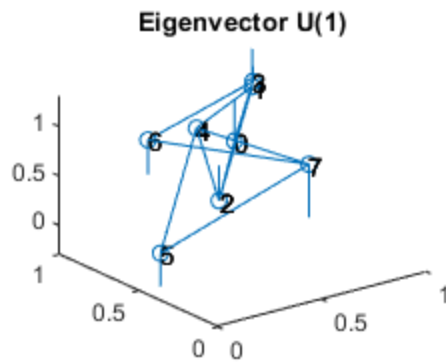
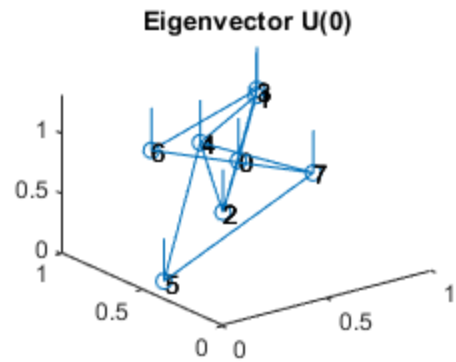
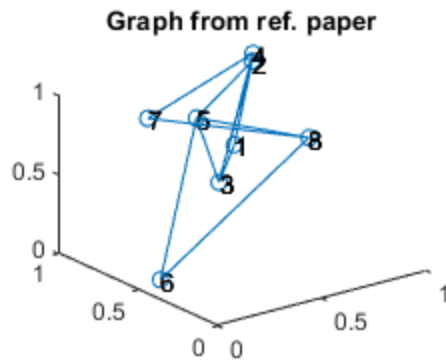
subplot(3,1,3);
plotLine3D(lineA,U2_59);
title('Eigenvector U(59)')

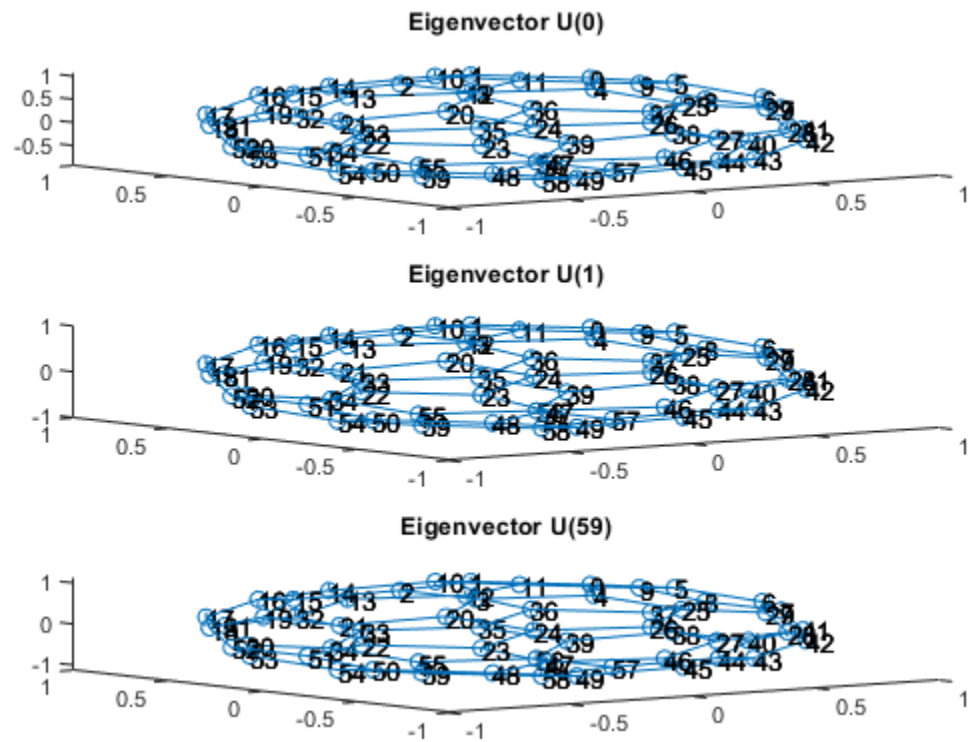
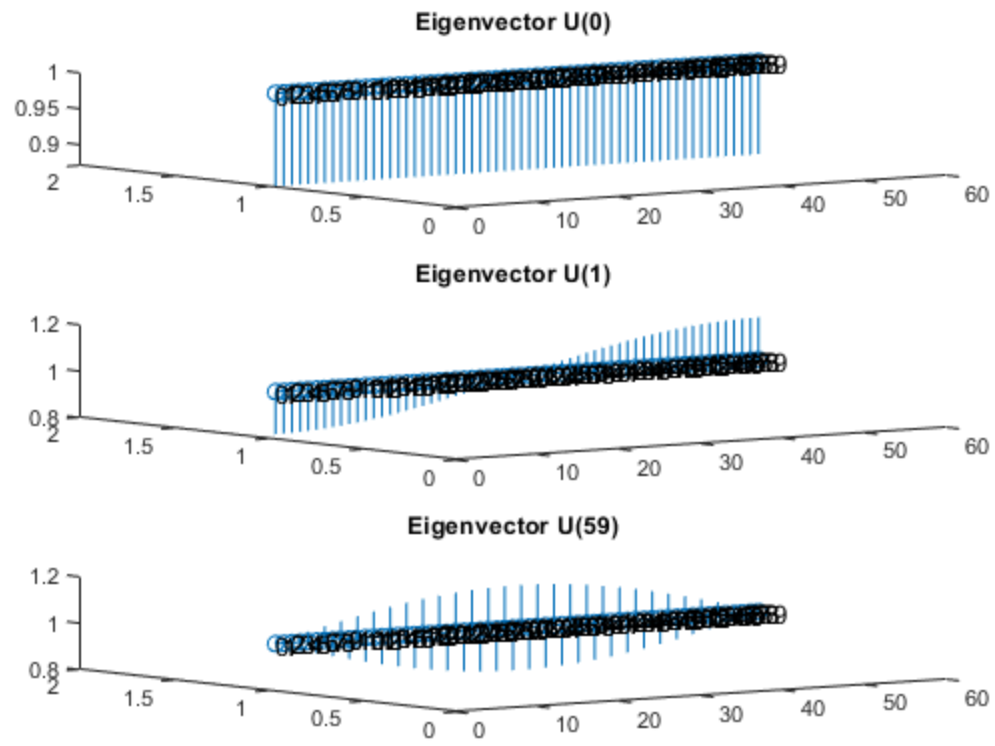
% Bucky Plots
figure;

subplot(3,1,1);
plot3Dv2(A2,xy,U3_0);
title('Eigenvector U(0)')
```

```
subplot(3,1,2);
plot3Dv2(A2,xy,U3_1);
title('Eigenvector U(1)')

subplot(3,1,3);
plot3Dv2(A2,xy,U3_59);
title('Eigenvector U(59)')
```





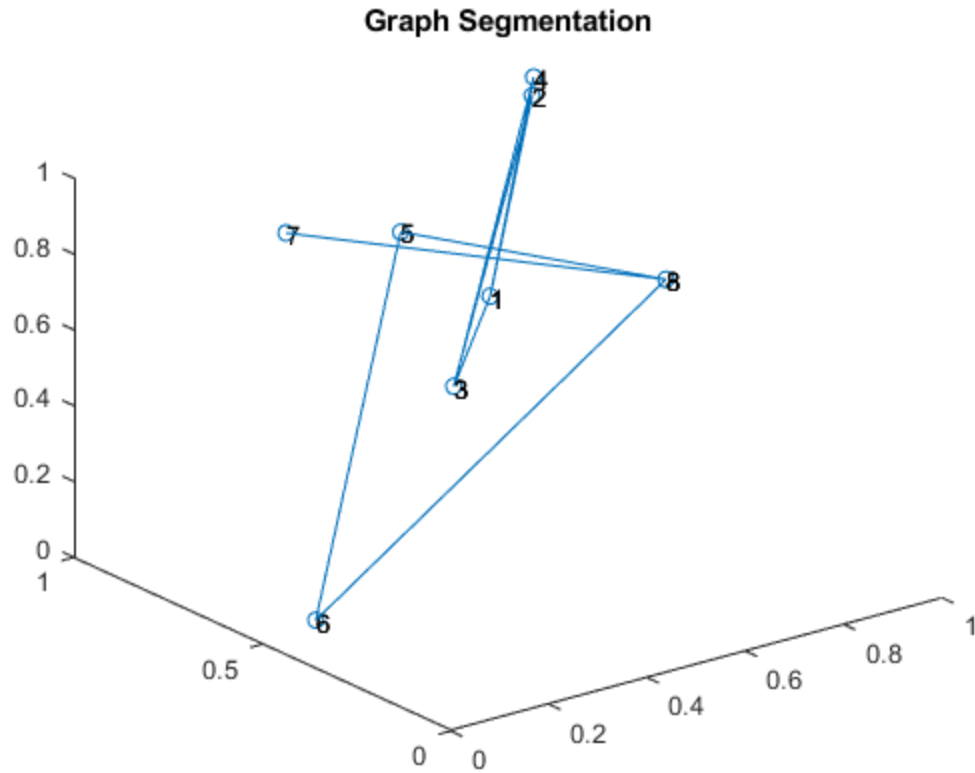
Question 2

```
S1 = spectralSimilarity(U1,2);
```

```
S2 = spectralSimilarity(U1,3);
```

Question 3

```
fiedler = U1(:,2);  
y = U1(:,2)/sqrt((U1(:,2).'*U1(:,2)));  
E = [];  
H = [];  
for i = 1:length(y)  
    if y(i) > 0  
        E = [E i];  
    else  
        H = [H i];  
    end  
end  
  
for i = 1: length(A)  
    for j = 1:length(A)  
        res1 = linearSearch(E,i);  
        res2 = linearSearch(H,j);  
        if (res1 == 0 && res2 == 0) || (res1 == 1 && res2 == 1)  
            A(i,j) = 0;  
        end  
    end  
end  
figure;  
plot3DGraph(A,randA)  
title('Graph Segmentation')
```



```
function S = spectralSimilarity(U,n)
S = [];
U = U(:,2:length(U));
visited = [];
for i = 1:length(U)
    Qi = U(i,1:n);
    for j = 1: length(U)
        current = [i j];
        [SearchResult,visited_op] = Search(current,visited);
        visited = visited_op;
        if i ~= j && SearchResult == 0
            Qj = U(j,1:n);
            Dij = sqrt((Qi - Qj)*(Qi - Qj).');
            S = [S; Dij];
        end
    end
end
end
end
```

```
function [op,visited] = Search(current,visited)
    if isempty(visited)
        visited = [visited; current];
```

```
        op = 0;
    else
        for i=1:size(visited,1)
            if (visited(i,1) == current(1,1) && visited(i,2) ==
current(1,2)) || (visited(i,2) == current(1,1) && visited(i,1) ==
current(1,2))
                op = 1;
                return
            end
        end
        visited = [visited; current];
        op = 0;
    end
end
```

```
function plotLine3D(A,val)
    cord = [];
    x = 1;
    y = 1;
    z = 1;
    for i=1:length(A)
        cord = [cord; x y z];
        x = x + 1;
    end
    plot3(cord(:,1),cord(:,2),cord(:,3),'o');
    for k = 1:length(A)
        for i = 1:length(A)
            if A(k,i) ~= 0
                x = [cord(k,1) cord(i,1)];
                y = [cord(k,2) cord(i,2)];
                z = [cord(k,3) cord(i,3)];
                k1 = int2str(k-1);
                text(cord(k,1),cord(k,2),cord(k,3),k1);
                line(x,y,z);
            end
        end
        x1 = [cord(k,1) cord(k,1)];
        y1 = [cord(k,2) cord(k,2)];
        z1 = [cord(k,3) (cord(k,3) + val(k))];
        line(x1,y1,z1);
    end
end
```

```
function A = LineGraph(n)
first_row = [0 1 zeros(1,n-2)];
last_row = [zeros(1,n-2) 1 0];
A = [first_row];
for i=0:n-3
```

```
        next_row = [zeros(1,i) 1 0 1 zeros(1, n-3-i)];  
        A = [A; next_row];  
    end  
    A = [A;last_row];  
end
```

```
function y = linearSearch(X,ele)  
for i= 1:length(X)  
    if X(i) == ele  
        y = 1;  
        return;  
    end  
end  
y = 0;  
return;  
end
```

Published with MATLAB® R2018a