Università della Svizzera italiana | Institute of Computing CI

Numerical Computing **2023**

Student: Leonardo Birindelli **Discussed with:** Morales Jeferson

Solution for Project 3

Due date: Wednesday, 8 November 2023, 11:59 PM

1. Spectral clustering of non-convex sets [50 points]

1. Run the Matlab function datasets/getPoints(). A graphical output of 6 different non-convex sets will be produced. Use this function to output the coordinate list of the different cloud-points and replace the variable Pts_dummy with the set you wish to cluster.

   Consider the set named half-kernel:can you identify the two obvious clusters in this dataset? Yes, I can , the two obviuous possible clusters are exactly the two archs of the graph. One cluster is the left arch and the other one is the right arch

   - Describe them briefly and explain what difficulties a clustering algorithm could eventually encounter in a scenario of this kind.

   The main difficulties that a clustering algorithm could encounter are strictly linked to the fact that it could not be able to correctly divide the structure in 2 different clusters. Indeed, the algorithm could understand to partition the structure not by the esplicity clearly identifiable of each cluster but trying to search for the nearest nodes (so the graph could be divide in 2 cluster by an horizontal line like it happens for the "Plot the K-means clusters")

2. Use the function minSpanTree() to find the minimal spanning tree of the full graph. Use this information to determine the ϵ factor for the ϵ-similarity graph.

   In order to determine the ϵ factor I executed the following matlab code:

```
% Create Gaussian similarity function
[S] = similarityfunc(Pts(:,1:2),exp(K)); % because K ~ log(n)

%it find the minimal spanning tree of the full graph
matrix_minimum_spanning_tree=minSpanTree(S);

% Obtain the epsilon the length of the longest edge in the minimum
    spanning tree

% Use max to find the longest edge in the minimum spanning tree of the
    full matrix
epsilon = max(matrix_minimum_spanning_tree,[],'all');
```

3. Complete the Matlab function epsilonSimGraph(). It should generate the similarity matrix of the ϵ-similarity graph

```
function [G] = epsilonSimGraph(epsilon,Pts)

n = size(Pts, 1);
G = zeros(n, n); % Initialize the similarity matrix with zeros
for i = 1:n
    for j = 1:n
        % Calculate the Euclidean distance between points i and j
        dist = norm(Pts(i, :) - Pts(j, :),2);

        % Check if the distance is less than epsilon
        if dist <= epsilon
            G(i, j) = 1;
            G(j, i) = 1;
        end
    end
end

end
```

4. Create the adjacency matrix for the ϵ similarity graph and visualize the resulting graph using the function gplotg().

```
% Create the epsilon similarity graph
[G] = epsilonSimGraph(epsilon,Pts);

%Creation of the adjacency matrix
W=S.*G;

%Graph visualization
figure;
gplotg(W,Pts(:,1:2))
title('Adjacency matrix for the epsilon similarity graph')
```

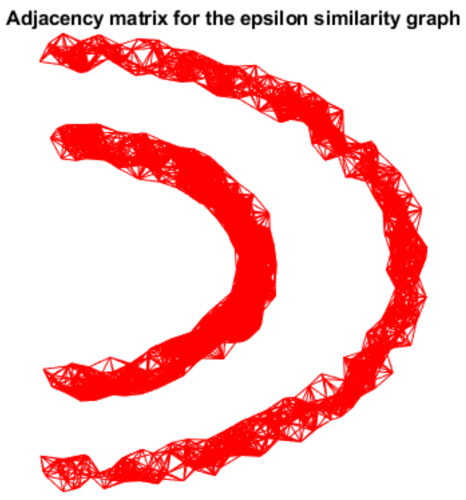

Figure 1: Half-kernel adjacency matrix image

5. Create the Laplacian matrix and implement spectral clustering. Your goal is to find the eigenvectors of the Laplacian corresponding to the K = 2 smallest eigenvalues. Afterwards, use the function kmeans_mod() to cluster the rows of these eigenvectors.

```
%Create the Laplacian matrix
[L,~] = CreateLapl(W);

% returns the K=2 smallest eigevalues , and their corrispective
    eigevectors , of the Laplacian matrix L
[V,~]=eigs(L,K,1e-6);

%spectral clustering
[D_spec, x_spec]= kmeans_mod(V,K,n);
```

6. Use the kmeans_mod() function to perform k-means clustering on the input points. Visualize the two clustering results using the function gplotmap(). You can debug your implementation by using as a reference the results reported in Figure 1, which are relative to the half-kernel dataset.

```
%K-means clustering
[D_kmeans, x_kmeans] = kmeans_mod(Pts(:,1:2),K,n);

%Create the Spectral clusters plot
figure;
subplot(1,2,1)
gplotmap(W,Pts(:,1:2),x_spec)
title('Plot the spectral clusters').

%Create the K-means clustering plot
subplot(1,2,2)
gplotmap(W,Pts(:,1:2),x_kmeans)
title('Plot the K-means clusters')
```

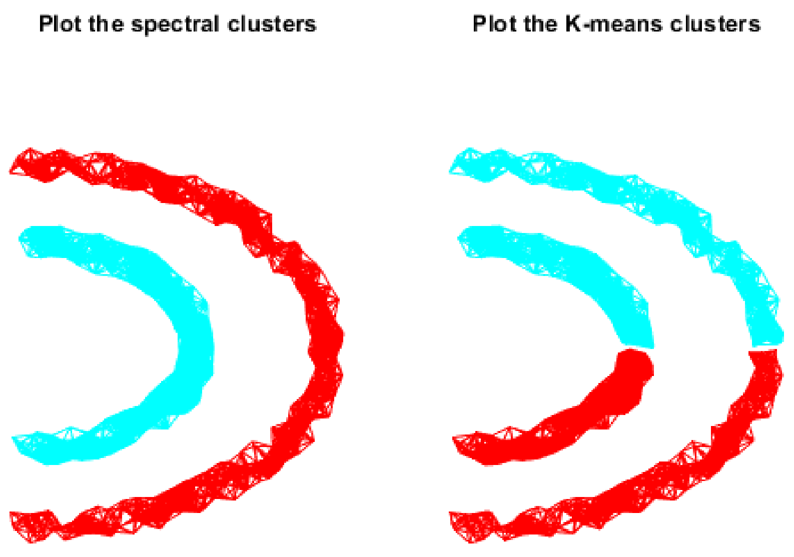

Figure 2: Half-kernel clustering results

7. Cluster the datasets i) Two spirals, ii) Cluster in cluster, and iii) Crescent & full moon in K = 2 clusters, and visualize the results obtained with spectral clustering and k-means directly on the input data. Do the same for i) Corners, and ii) Outlier for K = 4 clusters.

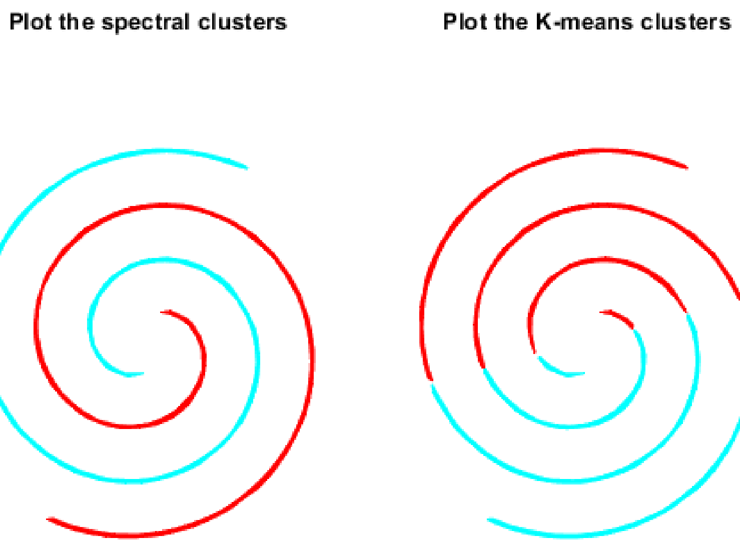

Figure 3: Two spirals clustering results

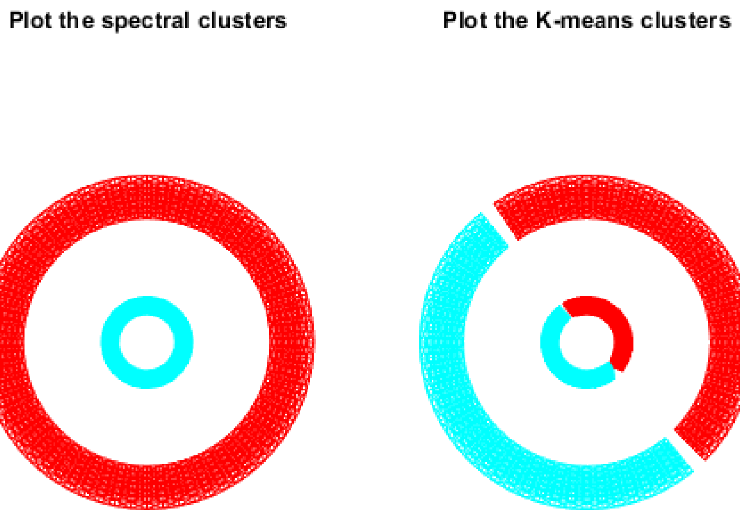

Figure 4: Cluster in cluster clustering results

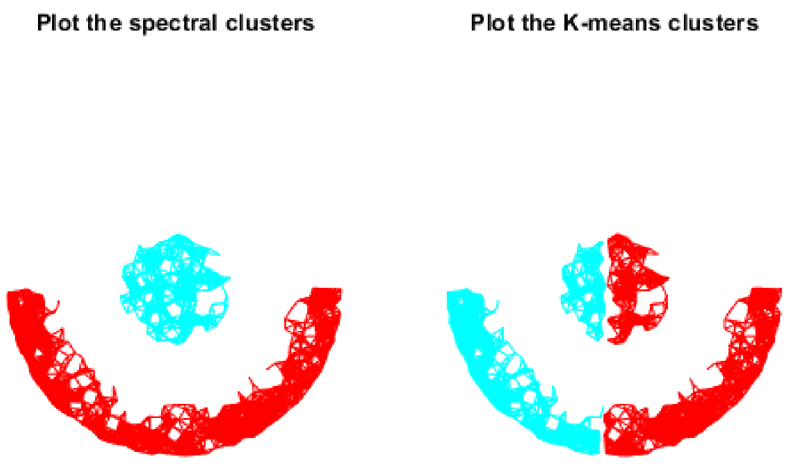

Figure 5: Crescent & Full moon clustering results

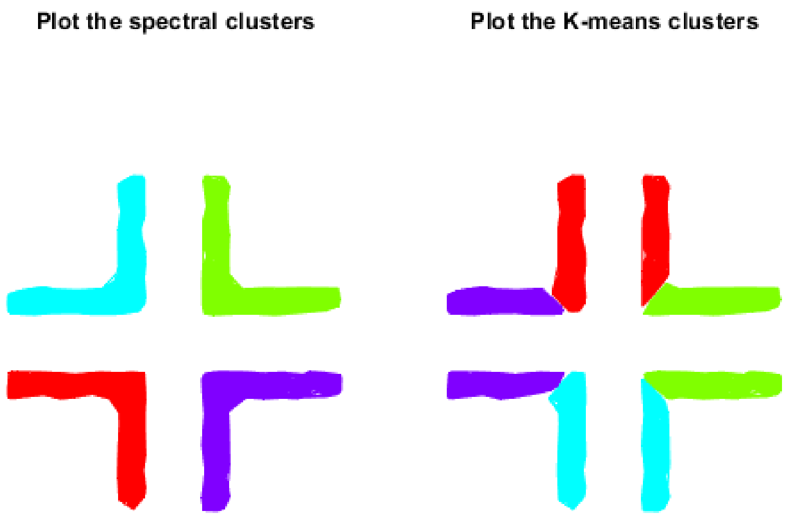

Figure 6: Corners clustering results

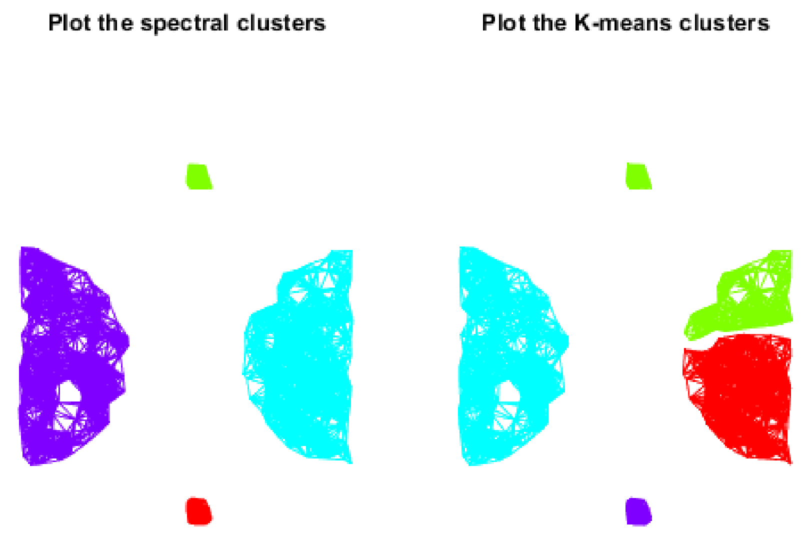

Figure 7: Outlier clustering results

I used the following code to generate plots

```
%% 1.7) Visualize spectral and k-means clustering results

K=2;
 datasets={twospirals(),
 clusterincluster(),
 crescentfullmoon(),
 corners(),
 outlier()};

 for i =1:numel(datasets)
     if(i==4)
         K=4;
     end

     Pts=datasets{i};
     n = size(Pts,1);

     [S] = similarityfunc(Pts(:,1:2),exp(K));
     matrix_minimum_spanning_tree=minSpanTree(S);

     epsilon = max(matrix_minimum_spanning_tree,[],"all");

     [G] = epsilonSimGraph(epsilon,Pts);

    W=S.*G;

    %Create the Laplacian matrix
     [L,~] = CreateLapl(W);
     [V,~]=eigs(L,K,1e-6);

     [D_spec,x_spec]= kmeans_mod(V,K,n);
    [D_kmeans,x_kmeans] = kmeans_mod(Pts(:,1:2),K,n);

    % the Spectral clusters plot
    figure;
    subplot(1,2,1)
    gplotmap(W,Pts(:,1:2),x_spec)
    title('Plot the spectral clusters');

    %the K-means clustering plot
    subplot(1,2,2)
    gplotmap(W,Pts(:,1:2),x_kmeans)
    title('Plot the K-means clusters');

    pause;
    close all;
end
```

Note: When the Gaussian similarity function is called, I also passed the parameter "exp(K)" to that function is order to initialise the value of sigma for the creation of the full similarity matrix. This choice depends on the fact that during the creation of the "outlier" graph clustering, the structure appeared sparsed and during the nodes clustering operation did not occur correctly. Therefore, to solve the issue, I chose to use this value to initialise sigma because since it appeared from theory that the number of partitions k must take on a value similar to the logarithm of the number of nodes in the graph,so $K{\sim}log(n)$ (see slide 19 of presentation1),I thought of making the value of sigma no longer depend on the number of nodes n , hence $\sigma = log(n)$, but directly on the number of clusters k. I translated this so that $\sigma = e^K$ and this solution implies that the graphs will be denser and the final output will be as required

2. Spectral clustering of real-world graphs [35 points]

1. Construct the Laplacian matrix, and draw the graphs using the eigenvectors to supply coordinates. Locate vertex i at position:

   $$(x_i, y_i) = (v_2(i), v_3(i))$$

   where v_2,v_3 are the eigenvectors associated with the 2nd and 3rd smallest eigenvalues. Figure 2 illustrates these 2 ways of visualizing the airfoil1 graph. Plot your results for all the three additionally supplied meshes, grid2, barth, 3elt.

   Meshe plots are the following :

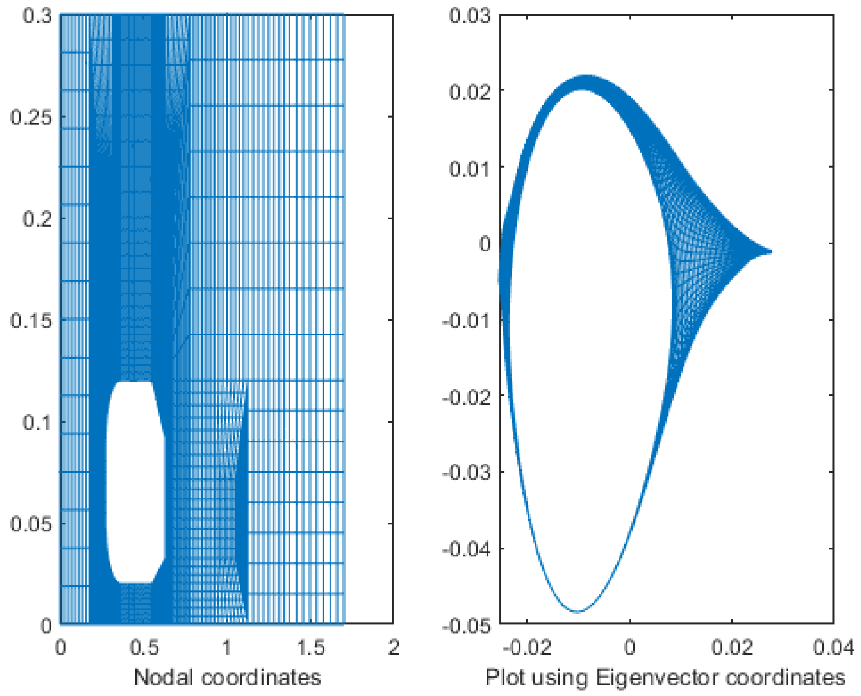

Figure 8: grid2 clustering results

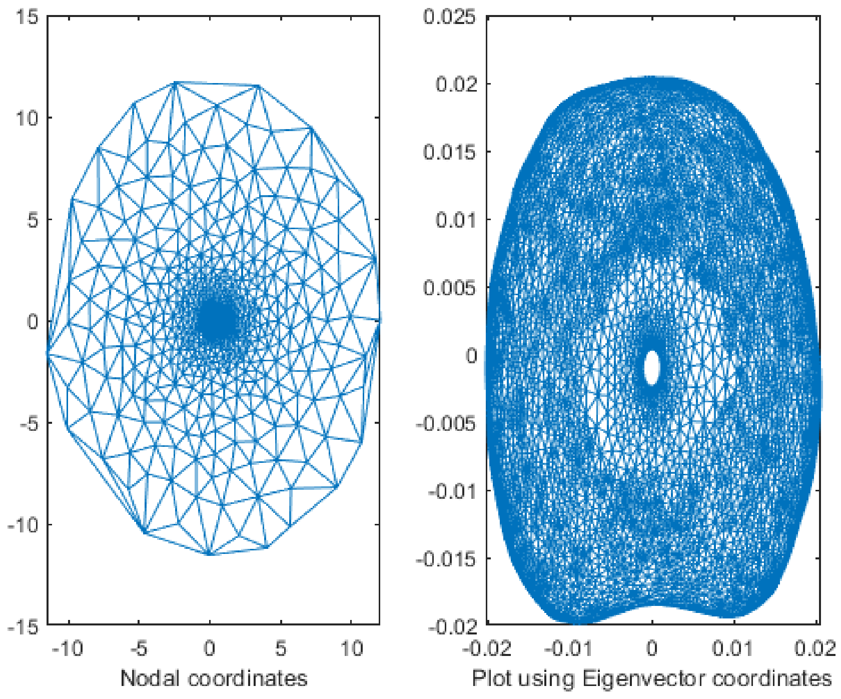

Figure 9: barth clustering results

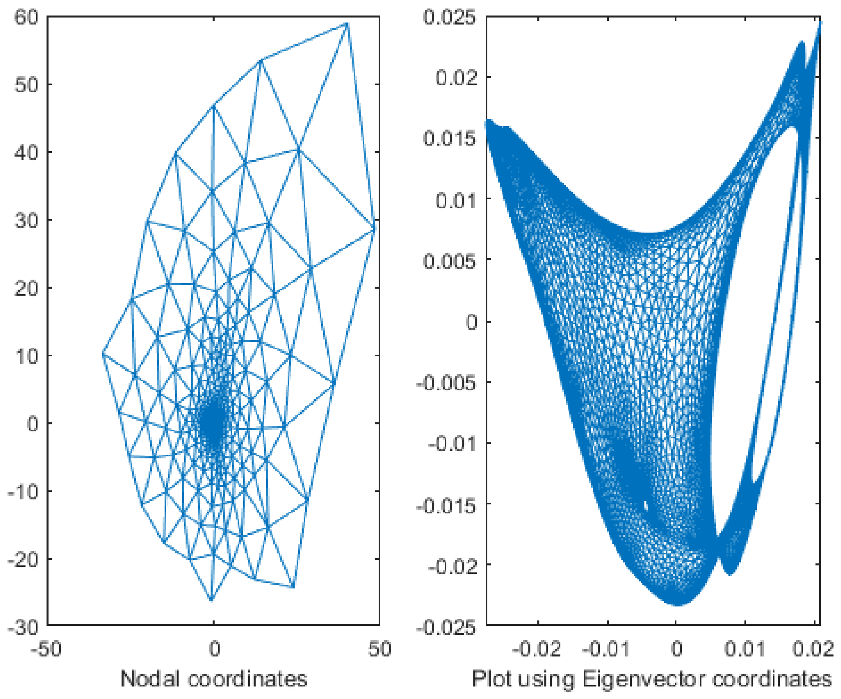

Figure 10: 3elt clustering results

2. Cluster each graph in K = 4 clusters with the spectral and the k-means method. Plot all of your results and describe the differences that you can see in the output of the 2 different algorithms and where they might come from. The clusters obtained by the 2 techniques in the case of the airfoil1 graph are presented in Figure 3.

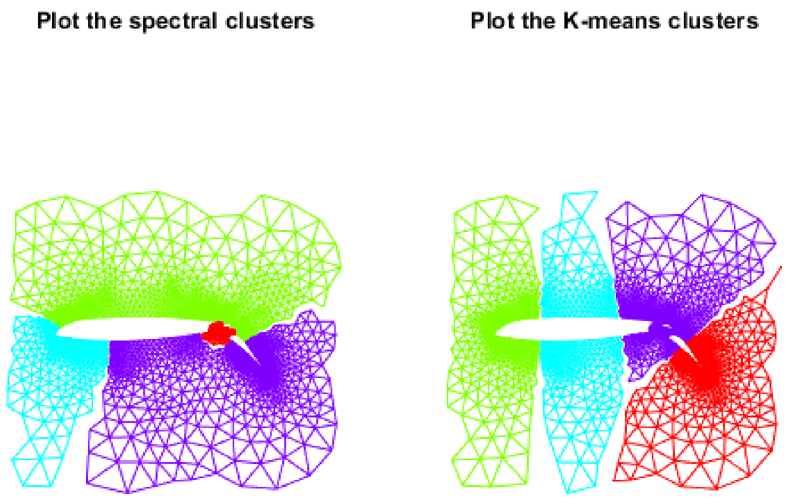

Figure 11: aifoil1 clustering results with K=4

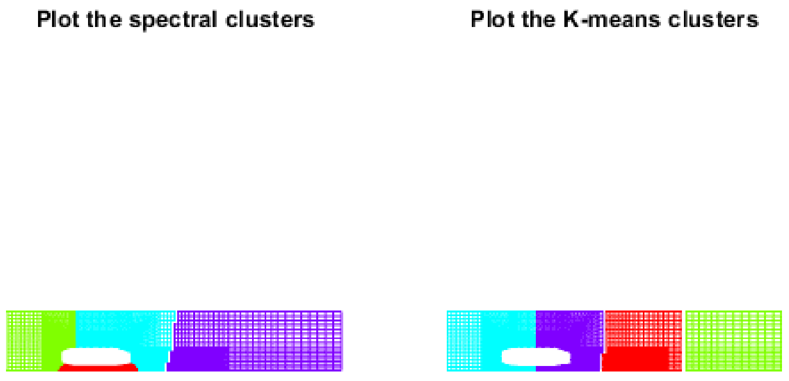

Figure 12: grid2 clustering results with K=4

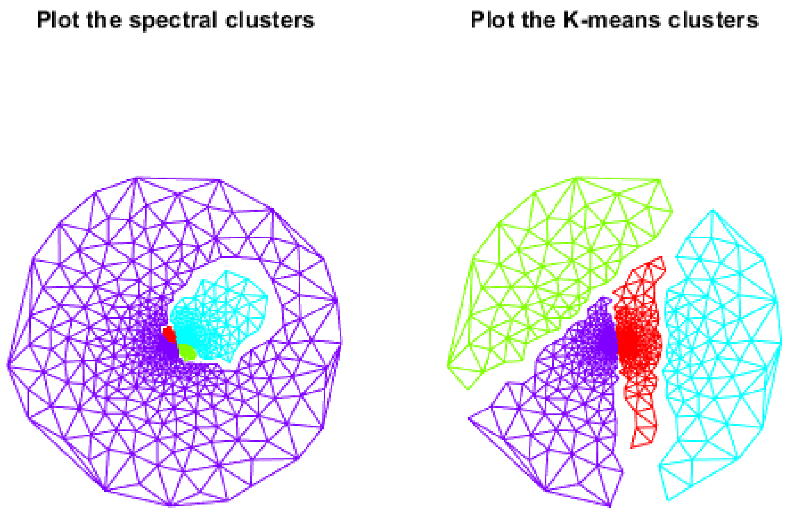

Figure 13: barth clustering results

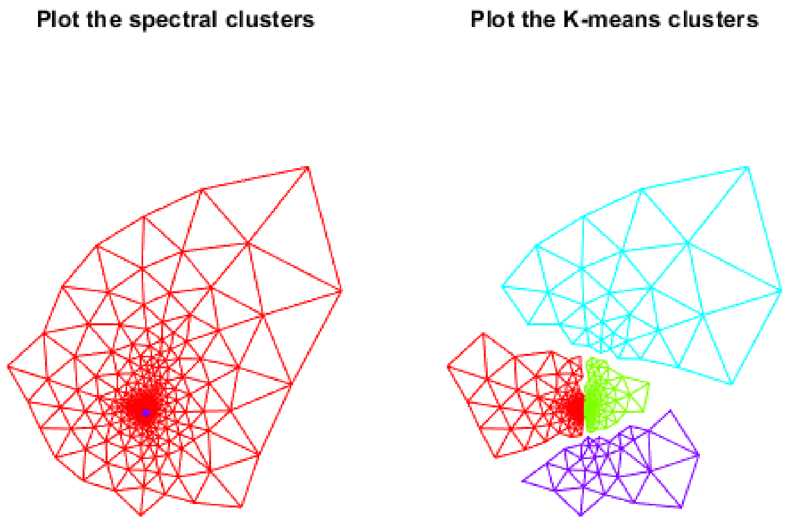

Figure 14: 3elt clustering results

Referring to the various graphs obtained, we can see substantial differences between spectral clustering and K-means clustering: on the one hand, with spectral clustering we are able to identify and distinguish groups of nodes on the basis of the characteristics that the structure assumes, and thus allow the creation of obvious clusters (see the 'airfoil' graph for example). While, on the other hand, K-means clustering is based on the idea of constructing clusters of nodes by minimising the distance of the points from the centroids of each cluster, which in practice it is translated into finding the optimal intermediate position between several nodes. However, this algorithm does not allow the characteristics of the nodes and thus the structure of the graph to be kept track of (e.g. see the 'airfoil' graph).

3. Report in Table 1 the number of nodes per cluster and plot a histogram of the reported values in a way similar to Figure 4. What can you observe?

Table 1: Clustering results, K = 4

Case	Spectral	K-Means
grid2	1305,785,827,379	238,1271,1183,604
barth	1601,1490,1405,2195	70,3526,70,3025
3elt	965,1794,1087,874	20,1768,2905,27

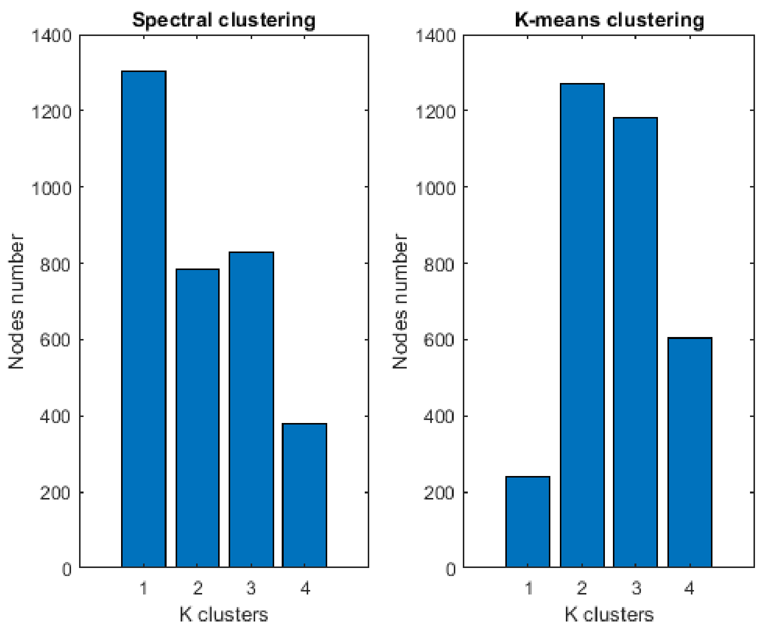

Figure 15: Histogram representing the number of nodes per cluster for the grid2 graph

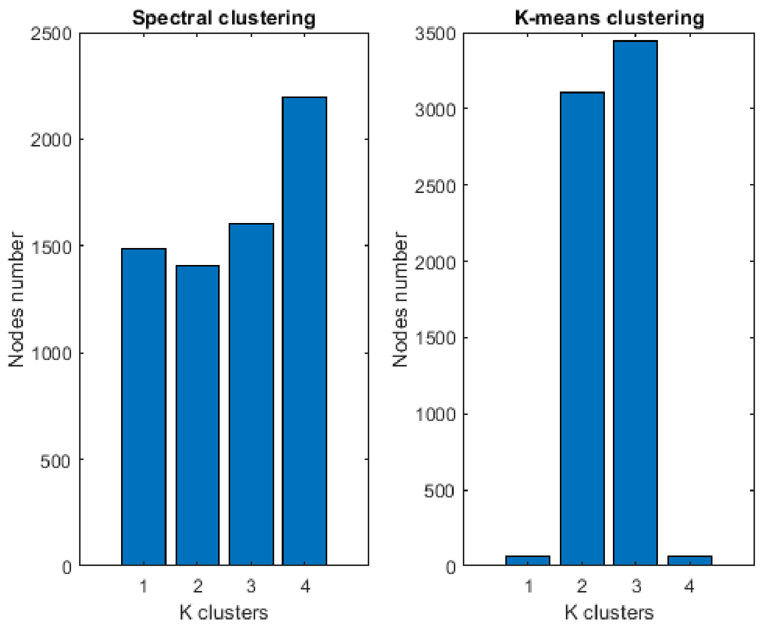

Figure 16: Histogram representing the number of nodes per cluster for the barth graph

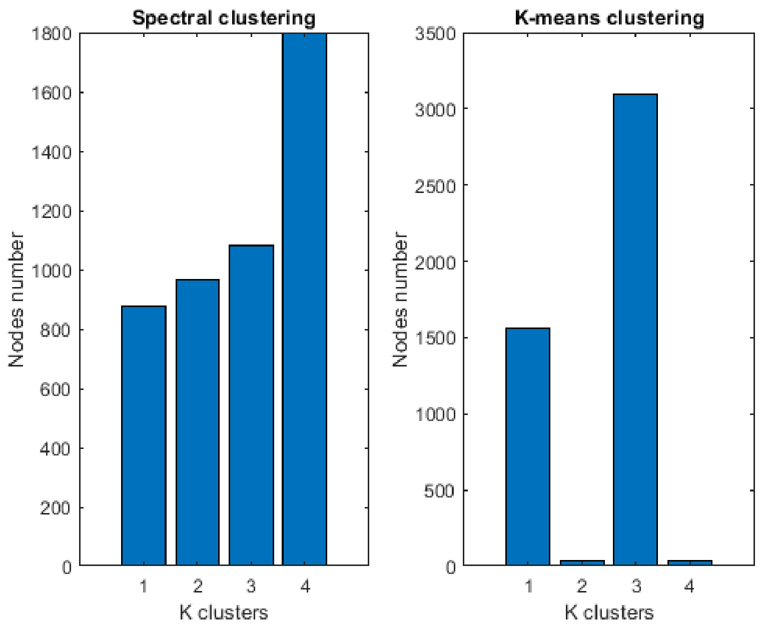

Figure 17: Histogram representing the number of nodes per cluster for the 3elt grah

Note : the code I used to generate the previous histograms is avaiable on the "ClusterGraphs.m" file

It is possible to notice that the nodes number in each cluster of the graph take on very different values when comparing the results of the two types of clustering, an example to analyze is how the number of nodes in cluster number one of the "barth" graph varies. In the case of spectral clustering, the number of nodes is 1305, whereas the result obtained by K-means is very small (only 20 nodes in the cluster).

3. References

1. ”Introduction to graph clustering” slides of the lesson of 18th October 2023 avaiable on iCorsi platform

2 Timothy A. Davis and Yifan Hu. The university of florida sparse matrix collection. ACM Trans. Math. Softw., (1):1:1–1:25, December 2011.

3 Ulrike Luxburg. A Tutorial on Spectral Clustering. Statistics and Computing, 17(4):395–416, December 2007.