Università della Svizzera italiana | Faculty of Informatics | Institute of Computing CI

Numerical Computing **2022**

Student: Kelvin Likollari Discussed with: Gerald Prendi, Johan Jacob

Solution for Project 4

Due date: Wednesday, 23 November 2022, 23:59 AM

Numerical Computing 2022 — Submission Instructions
(Please, notice that following instructions are mandatory: submissions that don't comply with, won't be considered)

- Assignments must be submitted to iCorsi (i.e. in electronic format).
- Provide both executable package and sources (e.g. C/C++ files, Julia). If you are using libraries, please add them in the file. Sources must be organized in directories called:
  Project_number_lastname_firstname
  and the file must be called:
  project_number_lastname_firstname.zip
  project_number_lastname_firstname.pdf
- The TAs will grade your project by reviewing your project write-up, and looking at the implementation you attempted, and benchmarking your code's performance.
- You are allowed to discuss all questions with anyone you like; however: (i) your submission must list anyone you discussed problems with and (ii) you must write up your submission independently.

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

(1) Compiling the file *getpoints.jl* yields a 6 non-convex and at the same time compact, set representations, which are all visualized and can be seen in the following image:

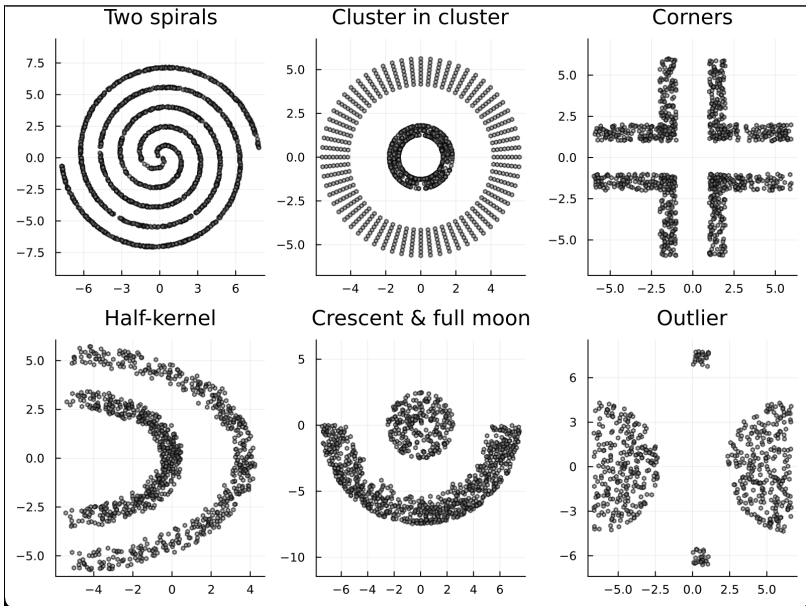


Figure 1: 6 non-convex set representations

The set two-spirals is visualized separately below:

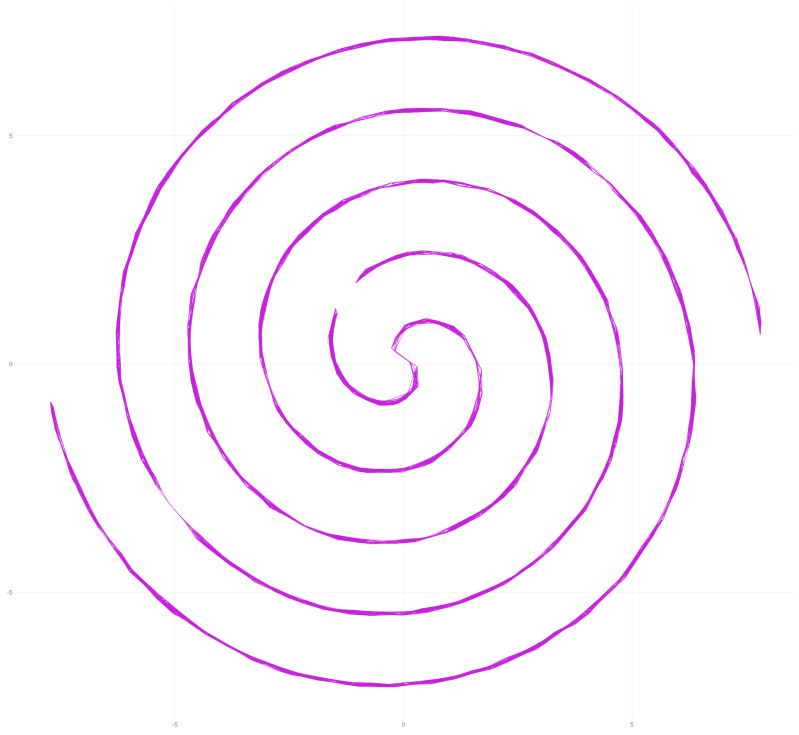


Figure 2: Two spiral

In Two Spirals, the two clusters visible are the points where the lines do not intersect. A clustering algorithm, given the above visualization, could encounter some problems. Effectively, the goal is to minimize the number of edge-cuts when partitioning the graph into 2 clusters. After plotting the two-spirals set representation, as seen in 2 it is evident that there is some point in its length where the two lines are not directly or indirectly connected. This provides a way to detect where the difficulties are. Using the k-means is not an optimal method since when applying the k-means method, the edge cut is increased to a great extent and this results in sub-optimality.

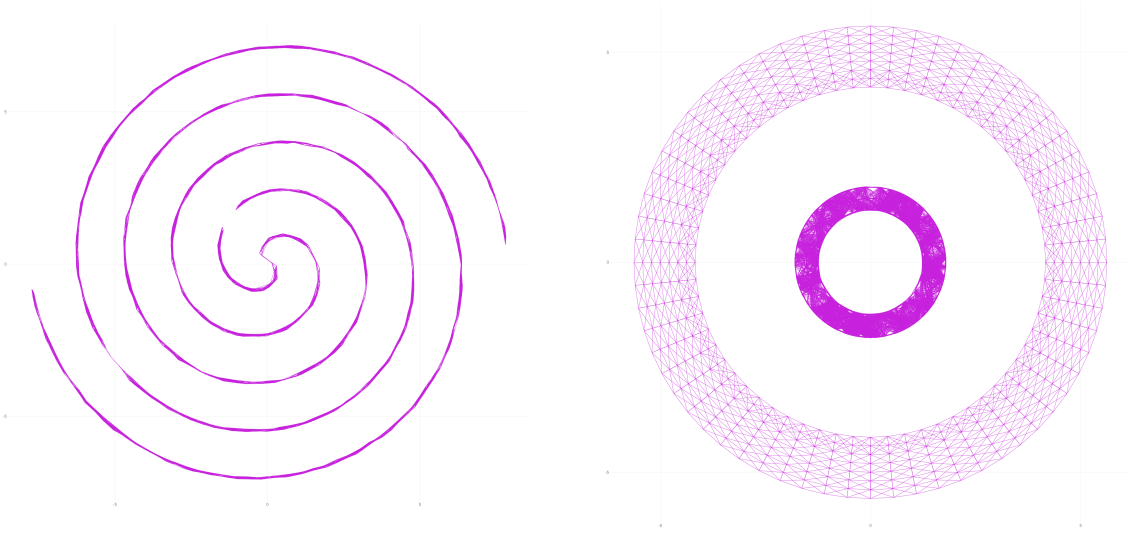
On the other hand, using the spectral method contradicts this sub-optimality since taking into account that the graph is actually split into 2 subgraphs, the clustering algorithm will start once from the left of the point where they stop spanning, and respectively once from the right. This way the entire graph will be spanned, with 2 edge cuts, and henceforth this proves that spectral clustering is more efficient than k-means clustering. The spectral clustering would have been even more efficient if there had not been a cut that prevents the connection of the two edge sets, because no cut is actually more performant than 1 cut.

(2) The function minspantree() has been used to determine the ϵ factor for the ϵ-similarity graph. ϵ was set to be the maximum edge weight in the adjacency matrix of the MST and has been computed as taking the maximum of the maximum of the columns of the adjacency matrix.

(3) For computing the ϵ-similarity graph, the respective Julia file was used. Effectively, the chunk of code used for implementing this function is given in the pdf (p.21). The given pseudocode has been translated into Julia code and can be found in the aforementioned Julia file. The idea behind its implementation is that after iterating over the rows and the columns, the distance between two vertices is computed in the first place. Following, the computed distance between those two vertices is compared to the previously-found epsilon value, where epsilon can be a number or a threshold, and in case it is less than epsilon, a matrix G is defined, where

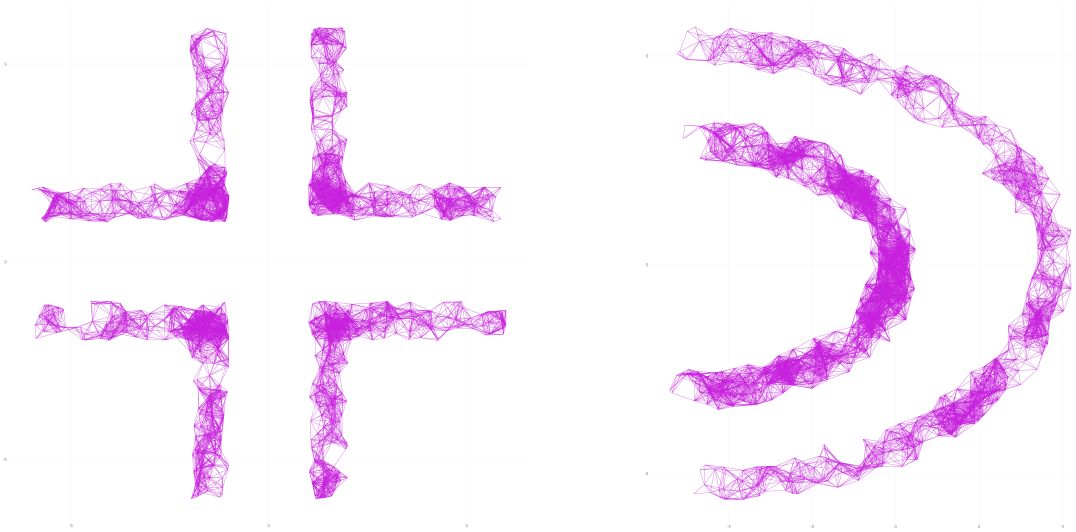
G(i, j) == G(j, i) == 1.

(4) The adjacency matrix for the ϵ-similarity graph has been created in the corresponding section in the Julia file. The results of plotting are visualized below:



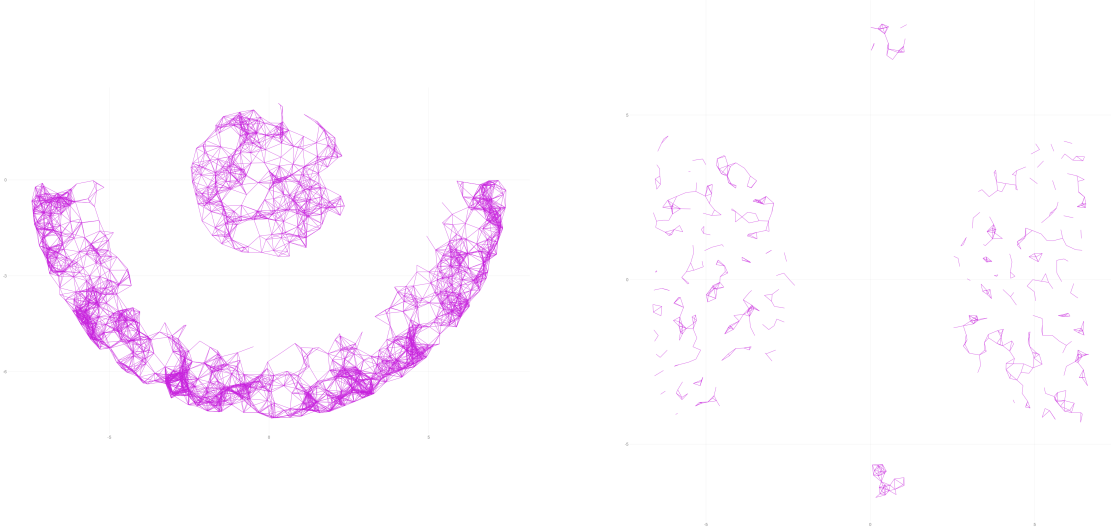
(a) Two spirals (b) Cluster in cluster

Figure 3: Two Spirals - Cluster in cluster



(a) Corners (b) Half-kernel

Figure 4: Corners & Half-kernel

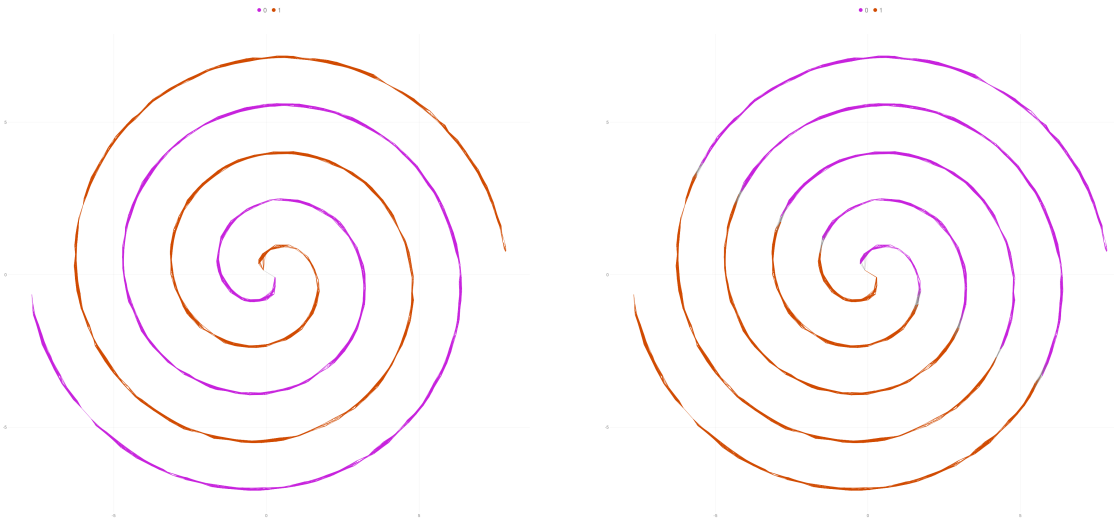


(a) Crescent & full moon (b) Outlier

Figure 5: Crescent & full moon - Outlier

(5) In order for the Laplacian Matrix to be created, a certain procedure has been followed. Essentially, after having calculated the similarity function, the MST and the ϵ factor alongside the ϵ-similarity graph, are all used so as to compute the adjacency matrix W_e. Afterwards, the eigenvalues are computed. Then, after the eigenvalues have been calculated, the eigenvectors are also calculated by assigning them a permutation vector (i.e. sorting them and permuting/reordering) which leads to getting the K = 2 smallest eigenvalues and later the clustering of their rows.

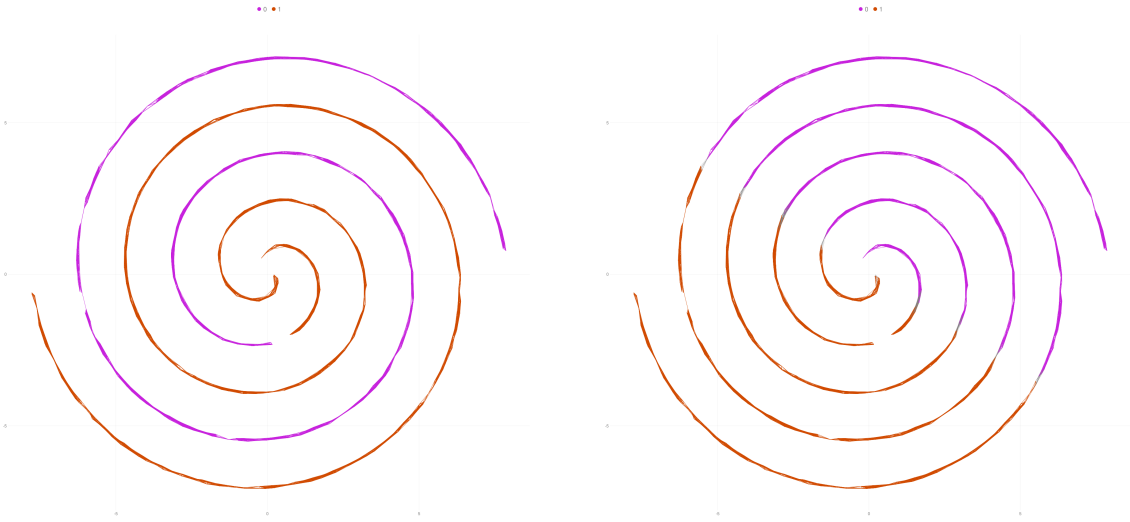
The following visualizations contain the clustered rows of the eigenvectors using the kmeans() method:



(a) Spectral Clustering (b) K-Means Clustering

Figure 6: Spectral & K-Means Clustering

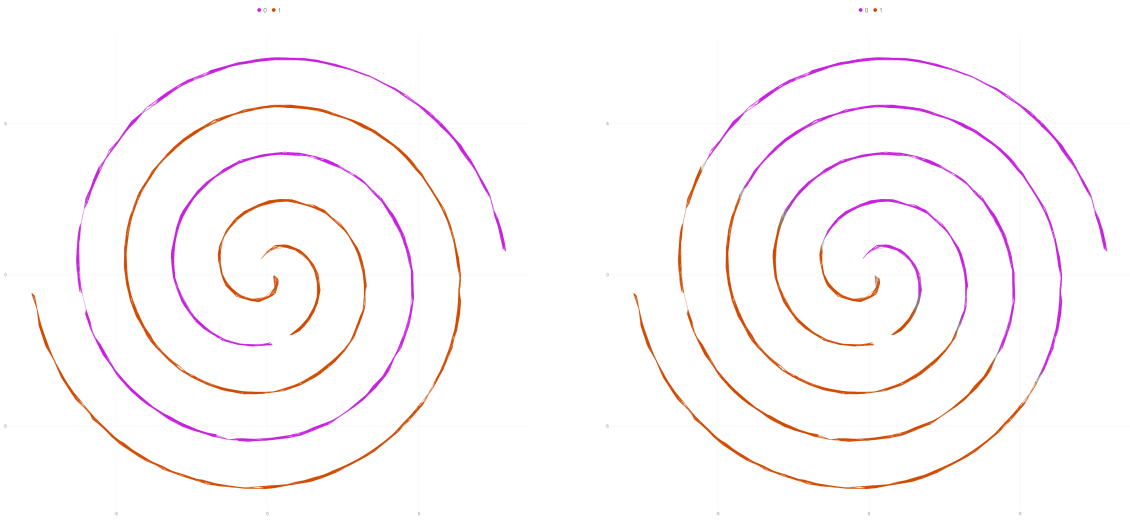
(6) The following sections contain the visualization when applying k-means clustering on the input points for all 6 datasets:



(a) Spectral Clustering (b) K-Means Clustering

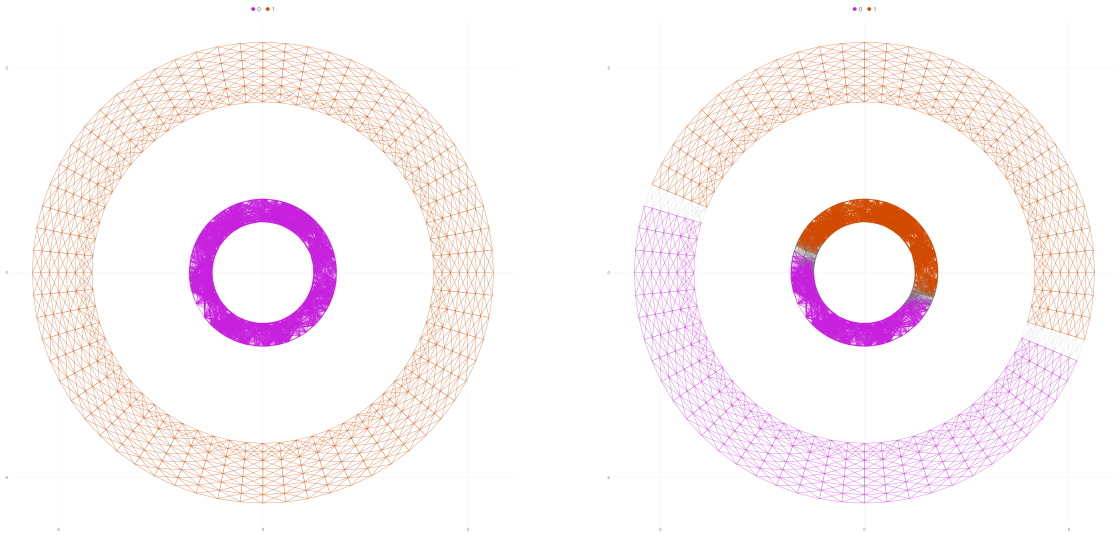
Figure 7: Spectral & K-Means Clustering - Two spiral

(7) The following section contains the datasets for Two spirals, Cluster in cluster, and Crescent & full moon for K = 2 clusters:



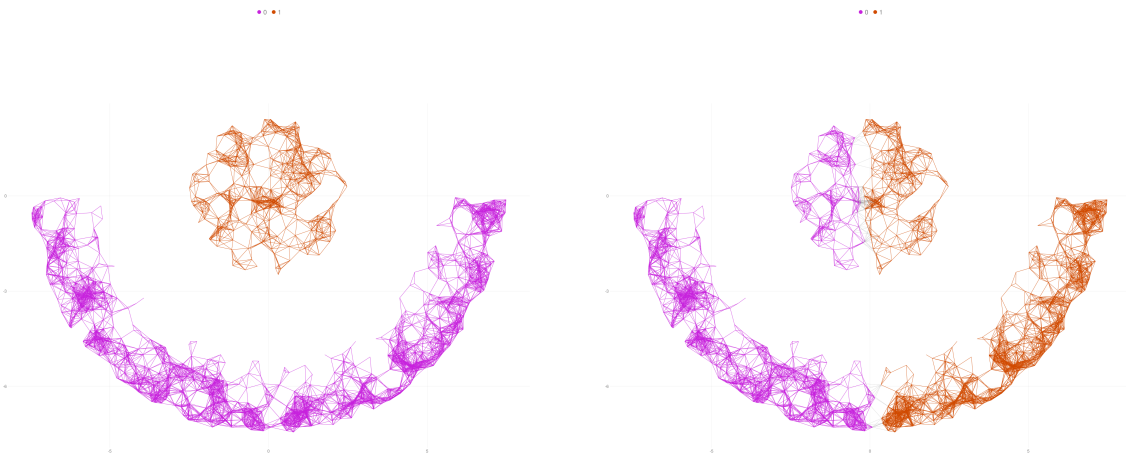
(a) Spectral Clustering (b) K-Means Clustering

Figure 8: Spectral & K-Means Clustering - Two spirals



(a) Spectral Clustering (b) K-Means Clustering

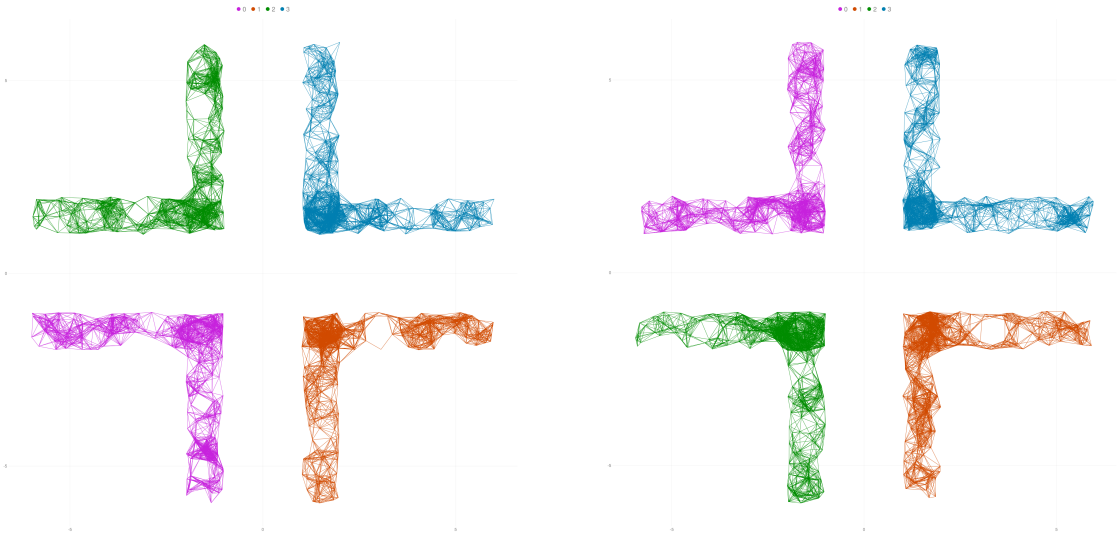
Figure 9: Spectral & K-Means Clustering - Cluster in cluster



(a) Spectral Clustering (b) K-Means Clustering

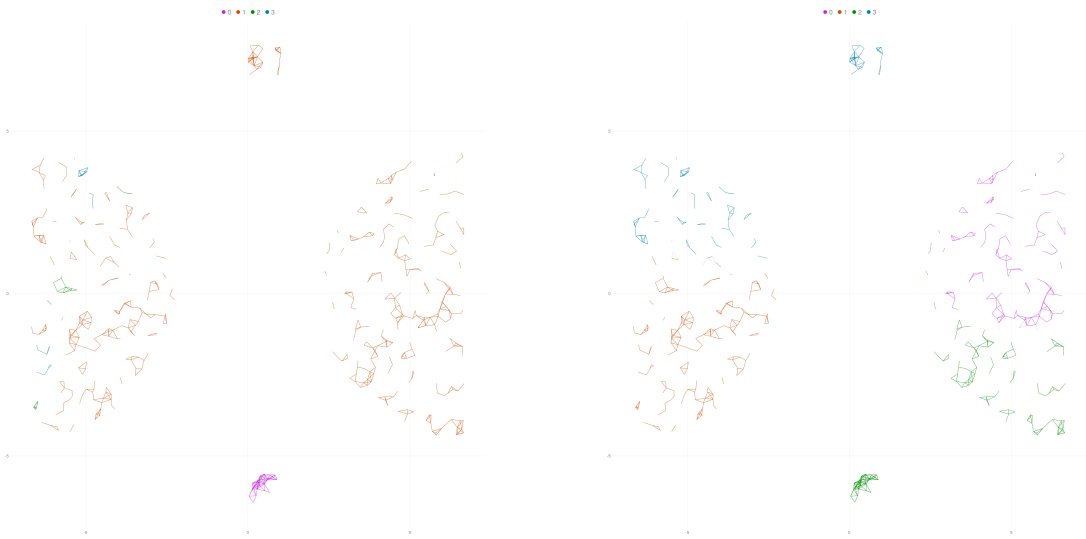
Figure 10: Spectral & K-Means Clustering - Crescent & full moon

For K = 4, the following graphs get plotted:



(a) Spectral Clustering (b) K-Means Clustering

Figure 11: Spectral & K-Means Clustering - Corners



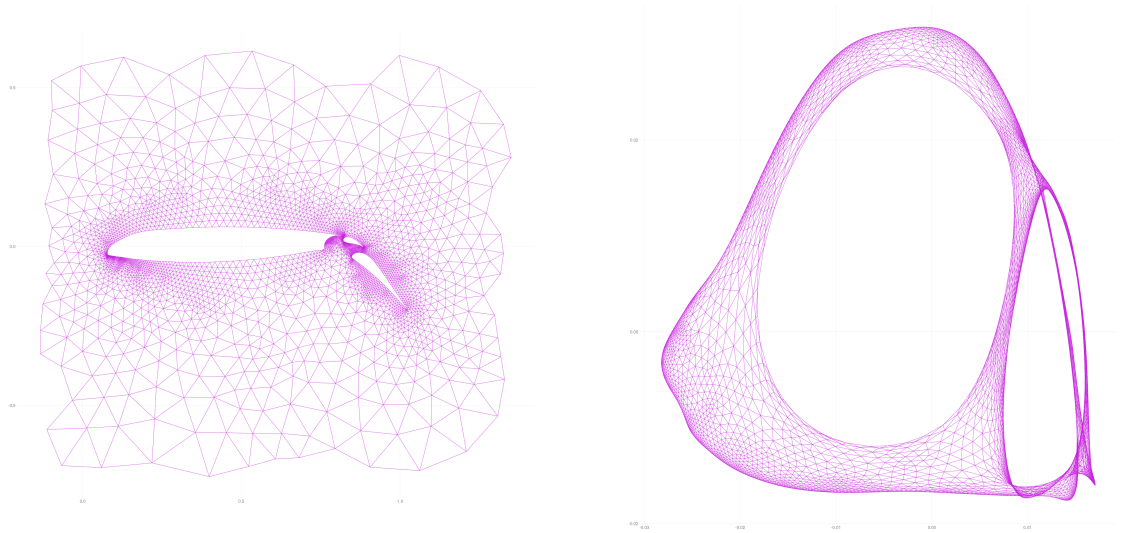
(a) Spectral Clustering (b) K-Means Clustering

Figure 12: Spectral & K-Means Clustering - Outliers

Note: variable σ was not used

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

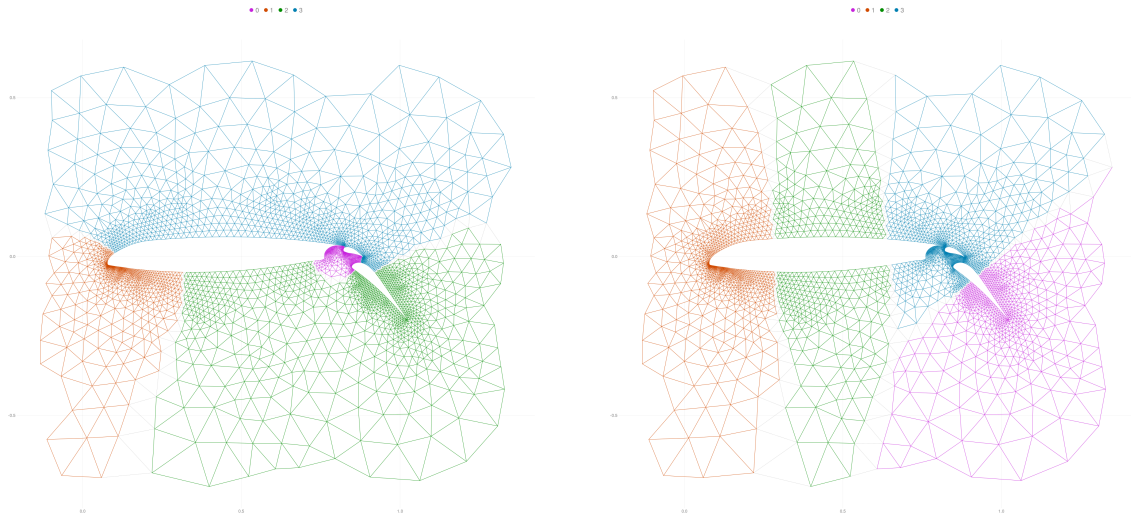
(1) The following figures illustrate 2 ways of visualizing the airfoil1 graph, using the second and third smallest eigenvalues:



(a) Airfoil1 Graph (1) (b) Airfoil1 Graph (2)

Figure 13: Airfoil Graphs

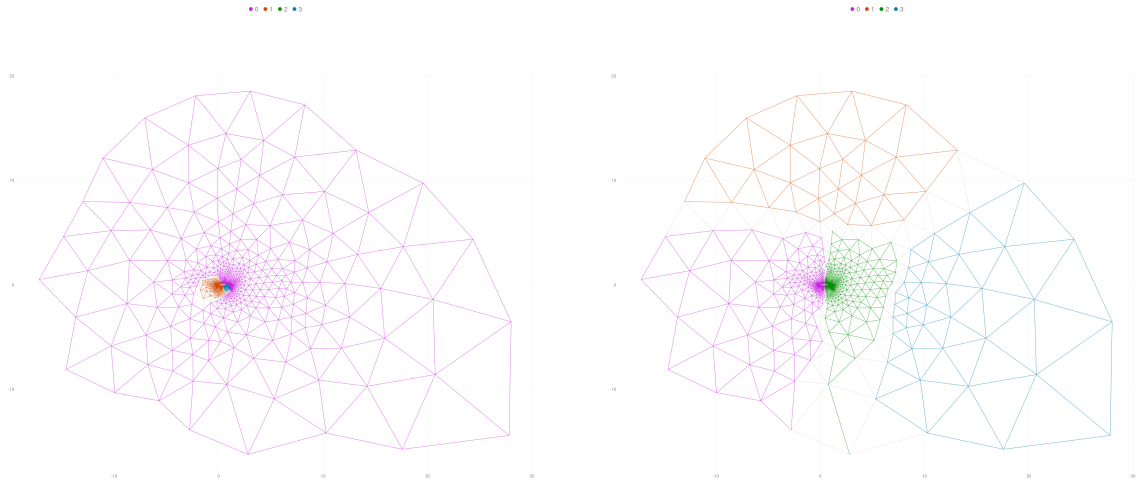
(2) Clustering each graph in K = 4 clusters with the spectral and the k-means method yields the following visualizations:



(a) Spectral Graph Clusters (1) (b) K-Means Graph Clusters (2)

Figure 14: Clusters in K = 4

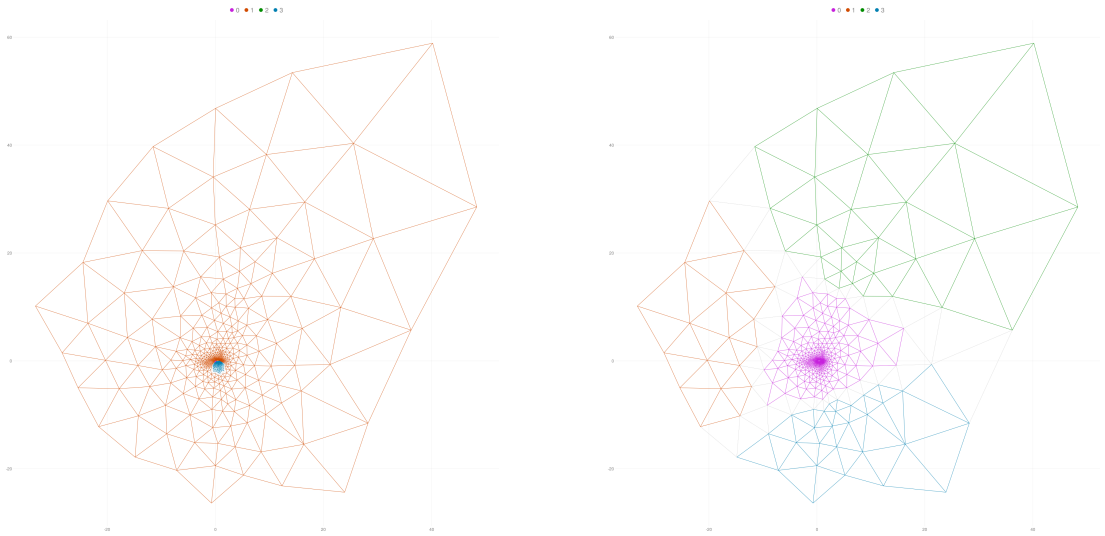
After visualizing the results for the *airfoil1* Graphs, the *barth4* results are obtained in a similar way, by uncommenting the respected file read in the Julia file. The results are shown below:



(a) Spectral Graph Clusters (b) K-Means Graph Clusters (2)

Figure 15: Barth4 Clusters

Lastly, the same procedure can be followed in order to generate the visualizations for the *3elt* graph:



(a) Spectral Graph Clusters (b) K-Means Graph Clusters (2)

Figure 16: 3elt Clusters

Looking at the aforementioned visualizations, there are multiple comments to be done. Specifically, one can clearly observe that in graphs depicted in 16, the two visualizations are severely identical in terms of node-positioning in the space. A sample difference that can be spotted is the (partial) alternation of color in different areas of the plot when toggling between Spectral and K-Means methods.

In general, the number of differences between the 2 clustering methods can be observed from the above-depicted graph visualizations. Effectively, starting with the *airfoil* clusters, one can clearly state that the density and the color of the nodes change significantly when toggling between the 2 clustering methods, although this is not of great importance. Moreover, taking into account the spectral clusters, it is evident that the nodes are symmetrical with respect to the center of the object.

Looking at the k-means clusters, one can see that the distance factor between the point and the object's center is taken into account. Effectively, the distance between the object's center point and the node is computed, and the least distance between the object's center point and the node is chosen, resulting in this node being mapped to the object's center point.

In terms of performance, as hinted from the above commentary, one can ambitiously claim that the spectral clusters method is more efficient compared to the k-means, if being applied to the *airfoil* graph. Since the spectral-clusters method takes into account symmetry, it is more efficient compared to using the k-means method and checking every single element (in this case node) that is closest to the centroid. This derivation can also be confirmed from the provided PDF stating that spectral clustering, compared to k-means is straightforward and the results obtained outperform the results obtained with other clustering methods.

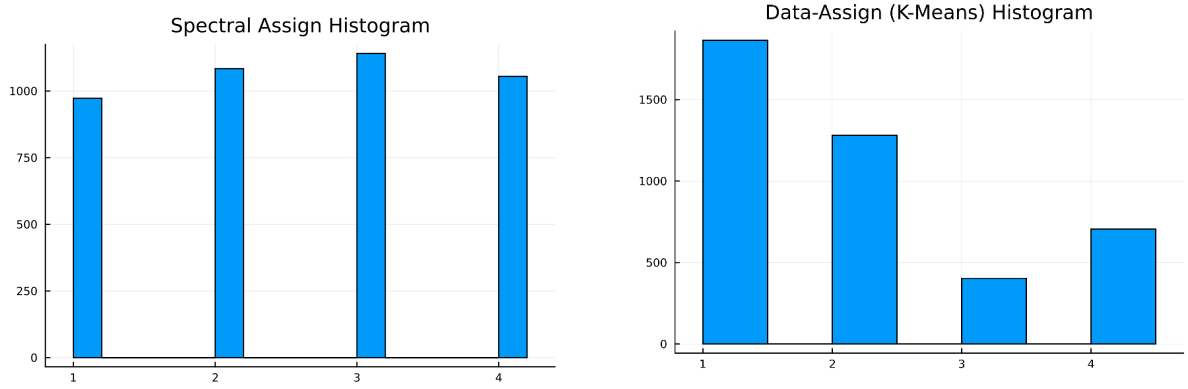
Observing the rest of the visualizations in 15 and 16, the conclusion is that they are quite equivalent. The extensive fill-in in each of the renderings is most visible in the graphs' centers. The nodes' positions in the space, on the other hand, are the same in both clustering approaches.

(3) The data containing the number of nodes per cluster have been stored in tabular view and are depicted below:

Case	Spectral	K-Means
grid2	973, 1084, 1055, 1141	1292, 1046, 412,703
barth	1326, 2171, 672, 1850	3635, 2314, 29, 41
3elt	1088, 965, 870, 1797	4642, 8, 30, 40

Table 1: Clustering Results, K = 4

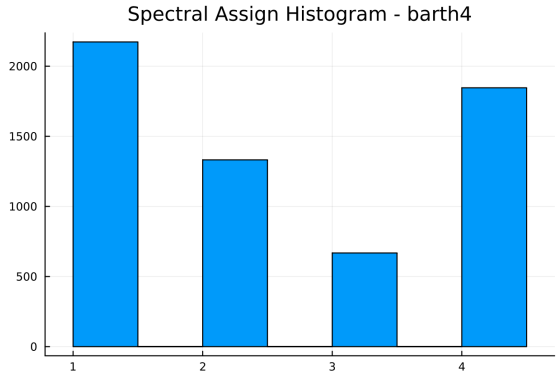
The Histograms representing the number of nodes per cluster for the *airfoil1 barth4* and *3elt* graphs are visualized in the following section:



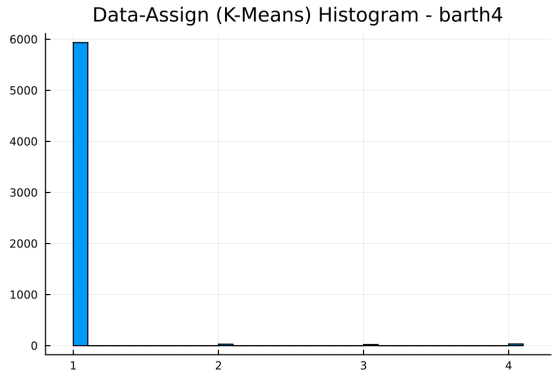
(a) Spectral Histogram

(b) Data-Assign (K Means) Histogram

Figure 17: Airfoil Histograms

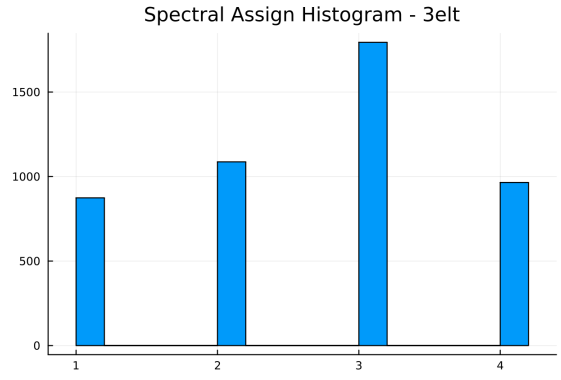


(a) Spectral Histogram

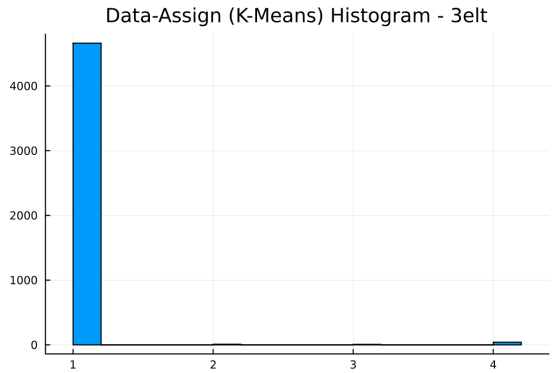


(b) Data-Assign (K Means) Histogram

Figure 18: Barth4 Histograms



(a) Spectral Histogram



(b) Data-Assign (K Means) Histogram

Figure 19: 3elt Histograms

One can draw some generalizations from the preceding histogram representations. There is no consistent pattern that such graphs follow (as spectral clustering does), but as previously said, it takes into account the symmetry of points along a certain viewpoint.

After employing the spectral clustering method and examining the clustering results table alongside the histograms, it is reasonable to conclude that there are no significant differences (or alternations) between the results obtained, in contrast to the k-means method, where the difference between particular points is overwhelming. This is impacted by the fill-in in the middle regions of the graph, since the more the fill-in, the greater the disparities in the visualization outputs of the associated histograms.