Università
della
Svizzera
italiana

Institute of
Computational
Science
ICS

Numerical Computing **2020**

Student: Claudio Maggioni Discussed with: –

Solution for Project 4

Due date: Wednesday, 18 November 2020, 11:55 PM

Numerical Computing 2020 — 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, Matlab). 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 [60 points]:

Plots for the *Two spirals*, *Cluster in cluster*, *Crescent moon*, *Full moon* clustering for $K = 2$ and for *Corners*, *Outlier* for $K = 4$ can be found in figures 2.2, 2.2, 2.2, 2.2, 2.2, and 2.2. All the plots are reproducible by simply running `ClusterPoints.m` once.

1.1. Observation on the *Spiral* set of points

It is possible to distinguish two distinct cluster in the *Spiral* set: if we consider this set of points as two intertwined non-intersecting spiral shaped curves, then each of the spiral can be considered as a cluster. However, it is possible that a naive clustering approach might not recognise these two clusters: since the spirals are intertwined and the points they are rotating on are close, averaging the points on each spiral leads to very close centroids and thus an algorithm heavily based on coordinate averaging (like k-means clustering) might have a hard time in identifying the spirals.

1.2. Choice of σ parameter for the Gaussian similarity function

According to the recommendations and rules of thumb on ϵ neighboorhood graph based spectral clustering, the σ parameter, as a rule of thumb, should be chosen in the order of $\log(n)$. However,

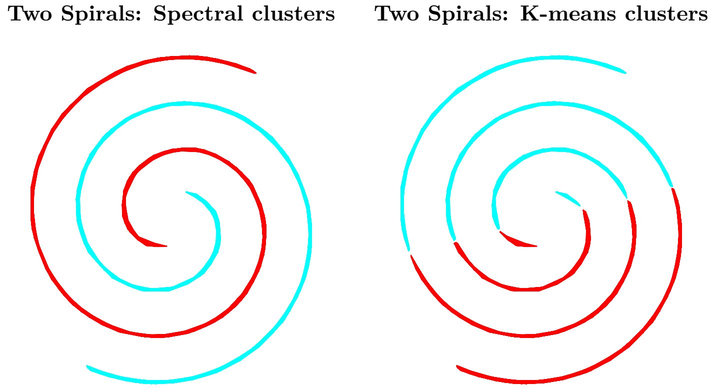

Figure 1: Spectral and k-means clustering graphs for *Two Spirals*

choosing $\sigma = \log(n)$ produces ill-conditioned i.e. (singular) Laplacian matrices for some graphs, which make spectral clustering inaccurate or impossible. Therefore, I have chosen $\sigma = 2\log(n)$. As the choice of this parameter is not governed by any strict law and this choice follows the rule of thumb and produces results that do not raise suspicion on incorrectness, I sticked to this choice.

2. Spectral clustering of real-world graphs [40 points]:

2.1. Plotting and commenting spectral and k-means clustering for several example graphs

Plots of spectral and K-means clustering for graphs *Airfoil*, *Barth*, *Grid2*, and *3elt* can be found respectively in figure 2.2, 2.2, 2.2 and 2.2. These graphs are reproducible by running `ClusterGraphs.m` once.

Overall, in all graphs spectral clustering seems to favour a more even distribution of vertices along the various clusters, while K-means generates clusters that cover similar areas. The extreme example of this is the *3elt* graph, where spectral and k-means clustering with wild imbalances respectively in cluster area and in cluster vertex count.

In the *Airfoil* graph it is unclear how the graph should be partitioned: both clusterings seem artificial and arbitrary. Again the comparison of cluster areas and cluster node counts follows what said above.

For the *Grid2* graph, spectral clustering seems to form a more natural cluster set by cutting somewhat radially along the center of the graph's "hole". In contrast, k-means clustering performs more artificial cuts along the y axis, creating clusters that resemble even slices of a cake.

Both *Barth* and *3elt* clusterings differ drammatically between spectral and k-means clustering and, albeit following the general observation stated above as well, both are not natural or obvious clusterings to human judgement.

2.2. Vertex count for spectral and k-means clustering

The table for the node counts of each cluster produced by spectral and k-means clustering of graphs *Airfoil*, *Grid2*, *Barth*, and *3elt* can be found in figure 2.2. The table and histograms found under the figures in the last section can be reproduced by running `ClusterGraphs.m` once.

As stated before, we observe that node counts for clusters generated by spectral clustering are more balanced with each other than the ones produced by K-means.

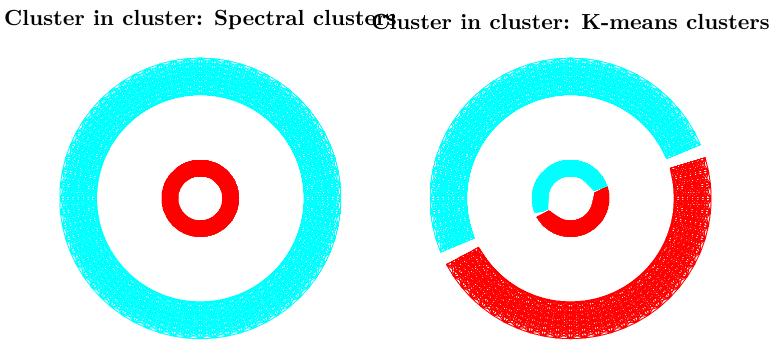

Figure 2: Spectral and k-means clustering graphs for *Cluster in cluster*

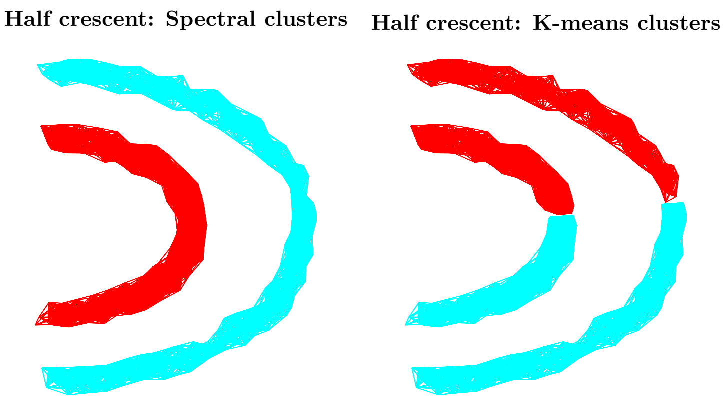

Figure 3: Spectral and k-means clustering graphs for *Half crescent*

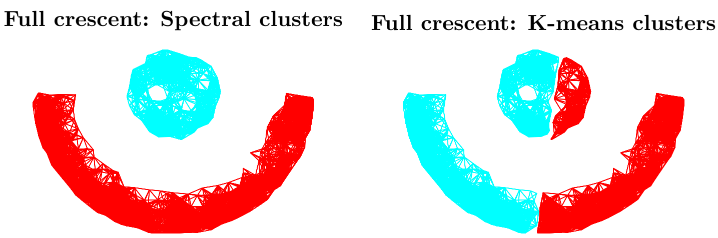

Figure 4: Spectral and k-means clustering graphs for *Full crescent*

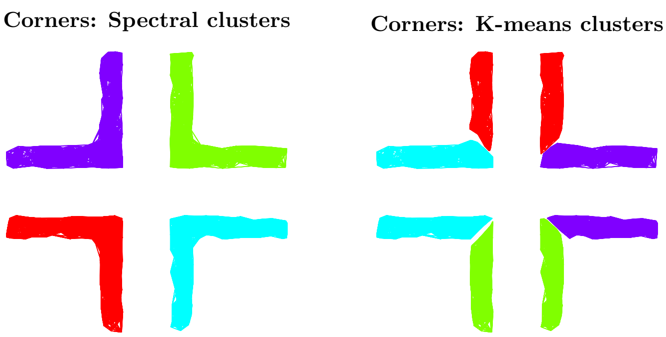

Figure 5: Spectral and k-means clustering graphs for *Corners*

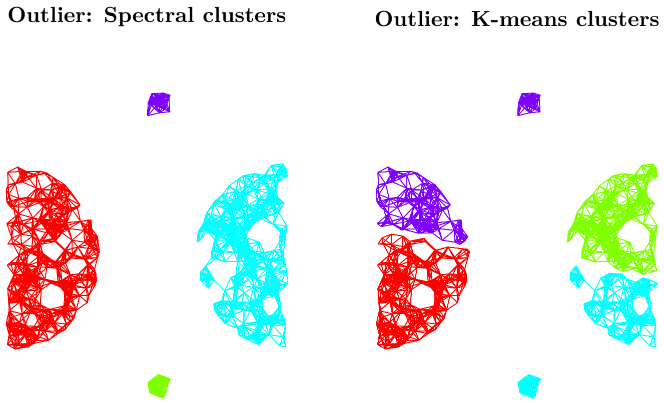

Figure 6: Spectral and k-means clustering graphs for *Outlier*

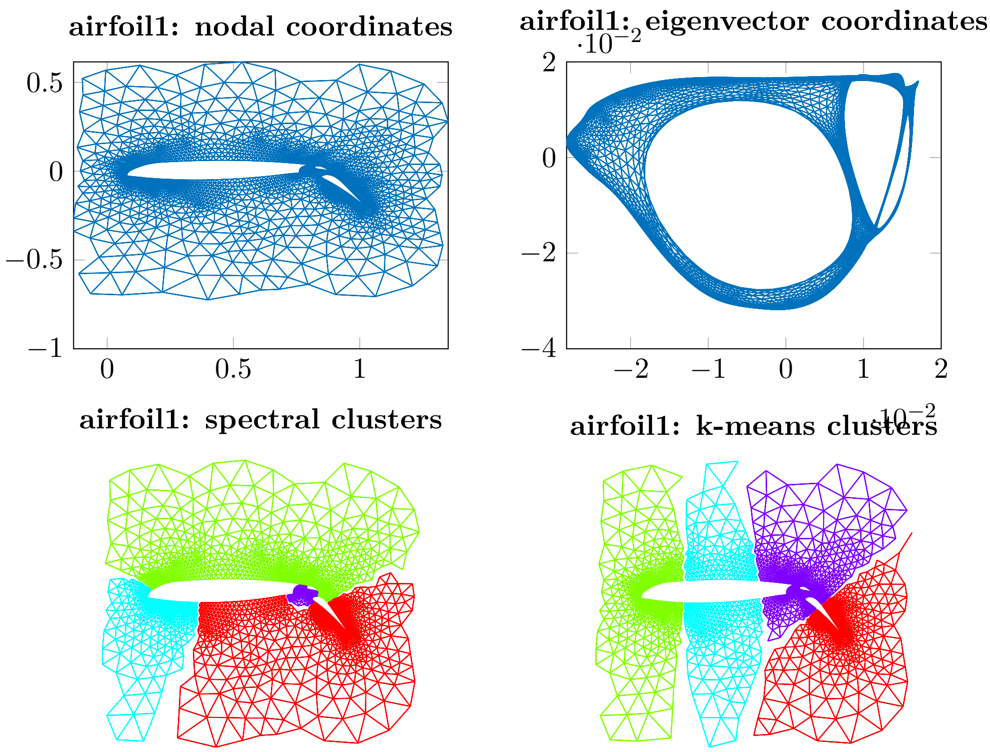

Figure 7: Graphs for *Airfoil1*

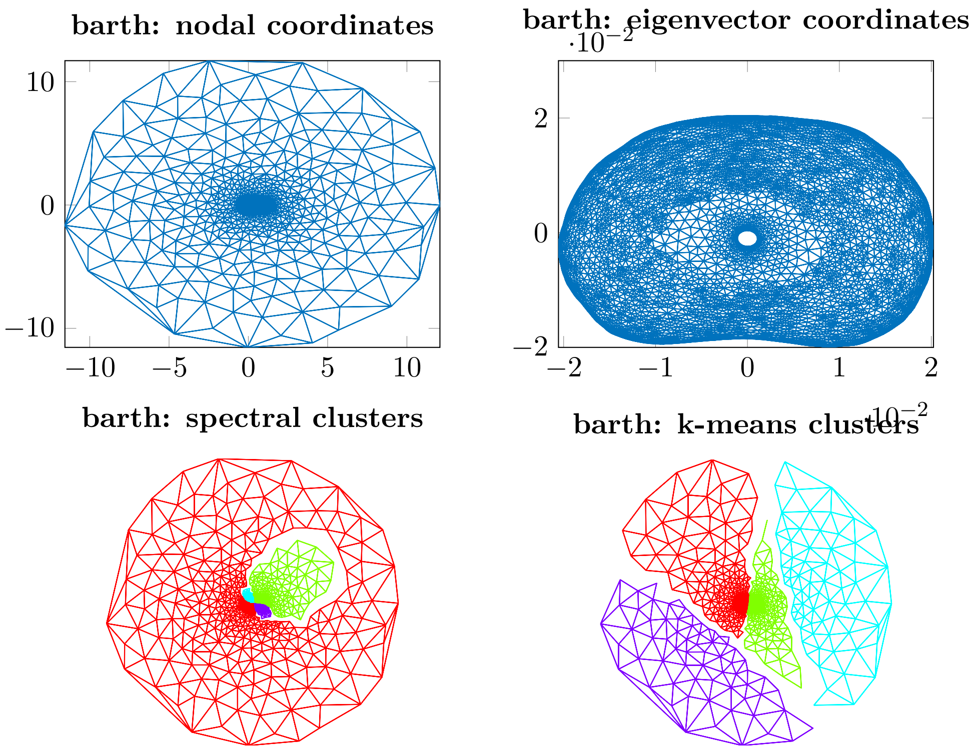

Figure 8: Graphs for *Barth*

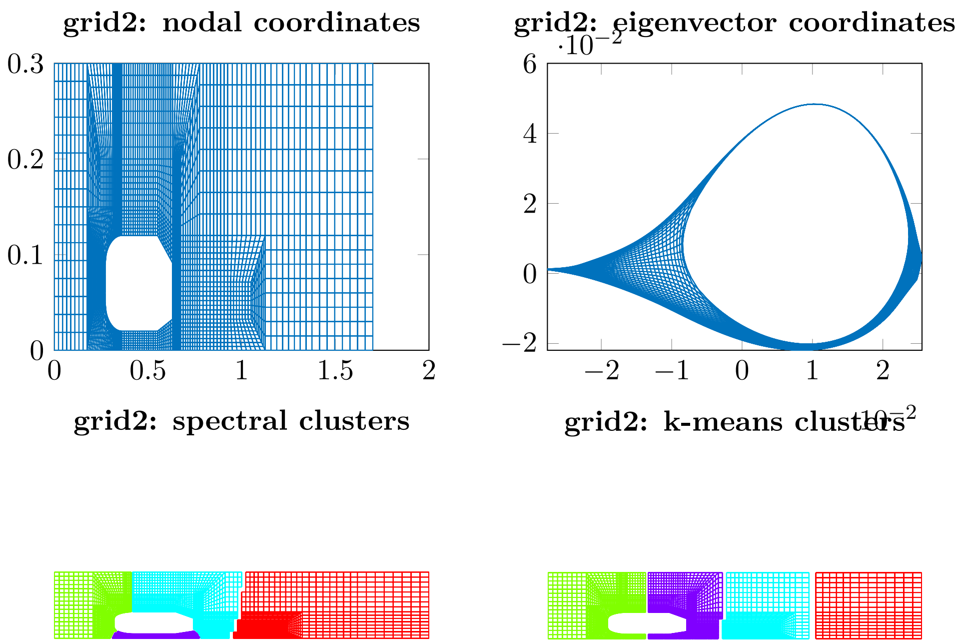

Figure 9: Graphs for *Grid2*

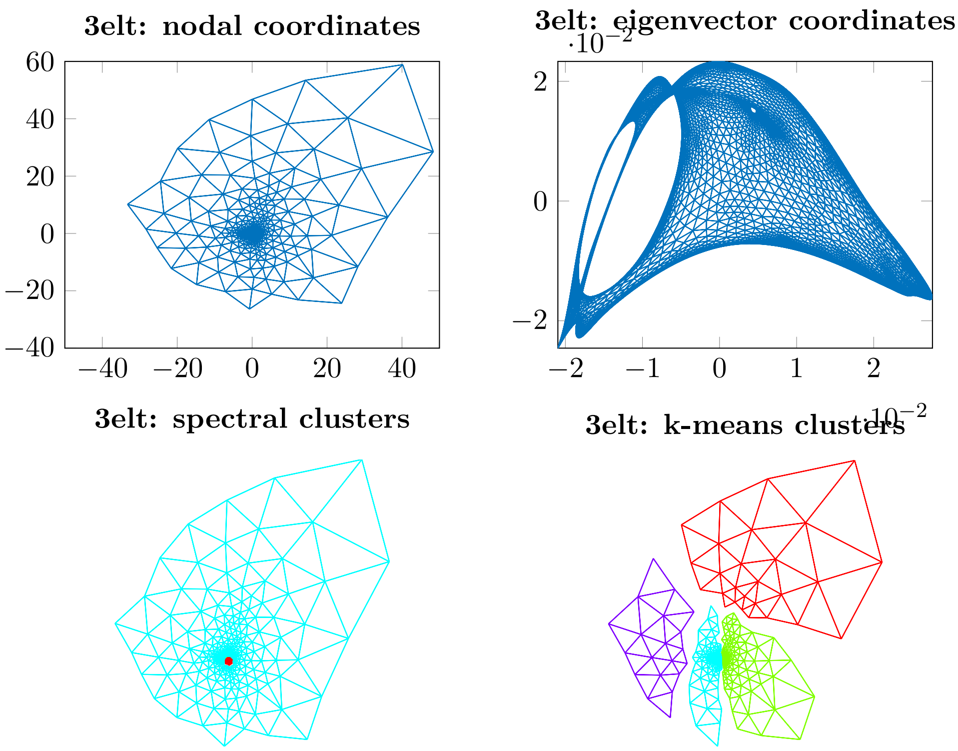

Figure 10: Graphs for *3elt*

Graph	Spectral				K-means			
	1	2	3	4	1	2	3	4
airfoil1	1150	1082	1050	971	1871	347	738	1297
barth	1601	1490	1405	2195	70	3526	70	3025
grid2	379	827	1305	785	1271	604	238	1183
3elt	965	874	1794	1087	13	1714	37	2956

Figure 11: Spectral and K-means clustering node counts for node counts