Università
della
Svizzera
italiana

Institute of
Computational
Science
ICS

High-Performance Computing Lab **2020**

Student: FULL NAME Discussed with: FULL NAME

Solution for Project 5

Due date: 11.11.2020 (midnight)

HPC Lab 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. Task: Install METIS 5.0.2, and the corresponding Matlab mex interface

2. Task: Construct adjacency matrices from connectivity data [10 points]

Run the Matlab script `src/read_csv_graphs.m` and complete the missing sections of the code. Your goal is to

- read the .csv files in MATLAB,
- construct the adjacency matrix $\mathbf{W} \in R^{n \times n}$ and the node coordinate list $C \in R^{n \times 2}$, where n is the number of nodes, and
- visualize the graphs using the function `src/Visualization/gplotg.m`

3. Task: Implement various graph partitioning algorithms [30 points]

- Run in Matlab the script `Bench_bisection.m` and familiarize yourself with the Matlab codes in the directory `Part_Toolbox`. An overview of all functions and scripts is offered in `Contents.m`.
- Implement **spectral graph bisection** based on the entries of the Fiedler eigenvector. Use the incomplete Matlab file `bisection_spectral.m` for your solution.
- Implement **inertial graph bisection**. For a graph with 2D coordinates, this inertial bisection constructs a line such that half the nodes are on one side of the line, and half are on the other. Use the incomplete Matlab file `bisection_inertial.m` for your solution.
- Report the bisection edgecut for all toy meshes that are either generated or loaded in the script
  "Bench_bisection.m." Use Table 1 to report these results.

Table 1: Bisection results

Mesh	Coordinate	Metis 5.0.2	Spectral	Inertial
mesh1e1	18			
mesh2e1	37			
netz4504_dual				
stufe				

4. Task: Recursively bisecting meshes [20 points]

The recursive bisecion algorithm is implemented in the file `rec_bisection.m` of the toolbox. Utilize this function within the script `Bench_rec_bisection.m` to recursively bisect the finite element meshes loaded within the script in 8 and 16 subgraphs. Use your inertial and spectral partitioning implementations, as well as the coordinate partitioning and the METIS bisection routine. Summarize your results in 2. Finally, visualize the results for $p = 16$ for the case "crack".

Table 2: Edge-cut results for recursive bi-partitioning.

Case	Spectral	Metis 5.0.2	Coordinate	Inertial
mesh3e1				
airfoil1				
3elt				
barth4				
crack				

5. Task: Comparing recursive bisection to direct k-way partitioning [10 points]

Use the incomplete `Bench_metis.m` for your implementation. Compare the cut obtained from Metis 5.0.2 after applying recursive bisection and direct multiway partitioning for the graphs in question. Consult the Metis manual, and type `help metismex` in your MATLAB command line to familiarize yourself with the way the Metis recursive and direct multiway partitioning functionalities should be invoked. Summarize your results in Table 3 for 16 and 32 partitions. Comment on your results. Was this behavior anticipated? Visualize the partitioning results for both graphs for 32 partitions.

Table 3: Comparing the number of cut edges for recursive bisection and direct multiway partitioning in Metis 5.0.2.

Partitions	Luxemburg	usroads-48	Greece	Switzerland	Vietnam	Norway	Russia
16							
32							

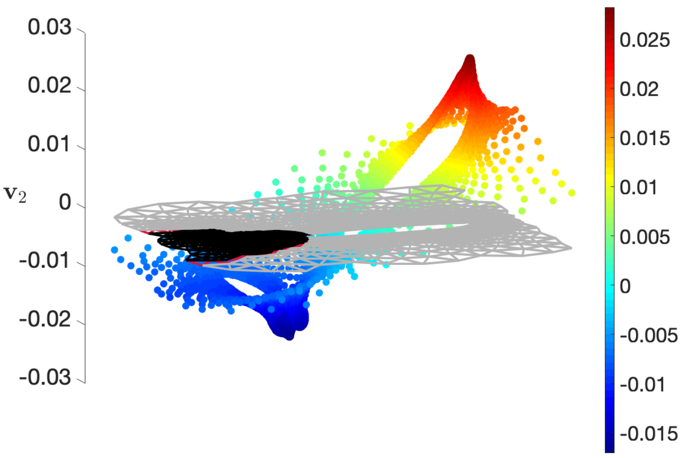

Figure 1: Partitioning the Airfoil graph based on the values of the Fiedler eigenvector. The two partitions are depicted in black and gray, while the cut edges in red respectively. The z-axis represents the value of the entries of the eigenvector.

6. Task: Utilizing graph eigenvectors [30 points]

Provide the following illustrative results. Use the incomplete script `Bench_eigen_plot.m` for your implementation.

1. Plot the entries of the eigenvectors associated with the first (λ_1) and second (λ_2) smallest eigenvalues of the graph Laplacian matrix $\mathbf{L}$ for the graph "airfoil1." Comment on the visual result. Is this behavior expected?

2. Plot the entries of the eigenvector associated with the second smallest eigenvalue λ_2 of the Graph Laplacian matrix $\mathbf{L}$. Project each solution on the coordinate system space of the following graphs: mesh3e1, barth4, 3elt, crack. An example is shown in Figure 1, for the graph "airfoil1".
Hint: You might have to modify the functions `gplotg.m` and `gplotpart.m` to get the desired result.

3. In this assignment we dealt exclusively with graphs $\mathcal{G}(V, E)$ that have coordinates associated with their nodes. This is, however, most commonly not the case when dealing with graphs, as they are in fact abstract structures, used for describing the relation E over a collection of entities V. These entities very often cannot be described in a Euclidean coordinate space. Therefore graph drawing is a tool to visualize relational information between nodes. The optimality of graph drawing is measured in terms of computation speed the ultimate usefulness of the resulting layout [1]. A successful layout should transmit the clearly the desired message, e.g the subsets of a partitioned graph. We will now see a spectral graph drawing method, which constructs the layout utilizing the eigenvectors of the graph Laplacian matrix $\mathbf{L}$. Draw the graphs mesh3e1, barth4, 3elt, crack, and their **spectral bi-partitioning** results using

the eigenvectors to supply coordinates. Locate vertex i at position:

$$x_i = (\mathbf{v}_2(i), \mathbf{v}_3(i)) \, ,$$

where $\mathbf{v}_2, \mathbf{v}_3$ are the eigenvectors associated with the 2nd and 3rd smallest eigenvalues of $\mathbf{L}$. Figure 2 illustrates these 2 ways of visualizing the partitions of the "airfoil1" graph.

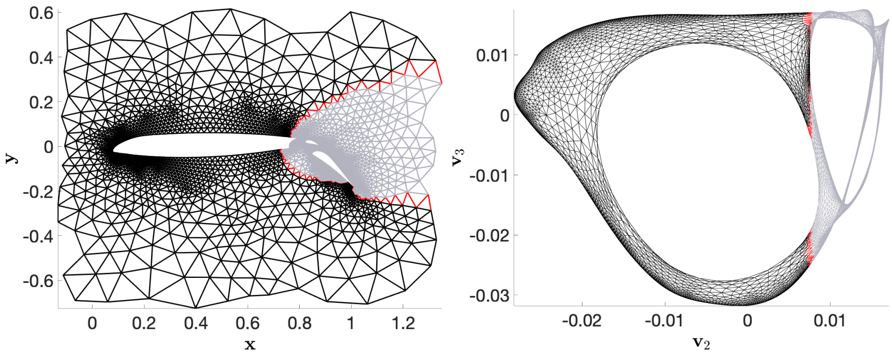

Figure 2: Visualizing the bipartitioning of the graph "airfoil1" with 4253 nodes and 12289 edges. Left: Spatial coordinates. Right: Spectral coordinates.

References

[1] Y. Koren. Drawing graphs by eigenvectors: Theory and practice. *Comput. Math. Appl.*, 49(11–12):1867–1888, June 2005.