Università
della
Svizzera
italiana

**Institute of
Computing
CI**

High-Performance Computing Lab **Institute of Computing**

Student: Paolo Deidda Discussed with: Lino Candian

Solution for Project 6

HPC Lab — 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.

Contents

1. Graph Partitioning with Matlab: Load balancing for HPC

2. Graph partitioning with Matlab: Exercises [85 points]

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

2.2. 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`

2.3. Task: Implement various graph partitioning algorithms [25 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				

2.4. Task: Recursively bisecting meshes [15 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".

2.5. 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 2: Edge-cut results for recursive bi-partitioning.

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

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							

2.6. Task: Utilizing graph eigenvectors [25 points]

In this section, we analyze the spectral properties of the graph Laplacian matrix L and their application to graph drawing and partitioning.

We computed the first two eigenvectors (v_1, v_2) associated with the smallest eigenvalues of the Laplacian for the `airfoil1` graph. The results are shown in Figure 1.

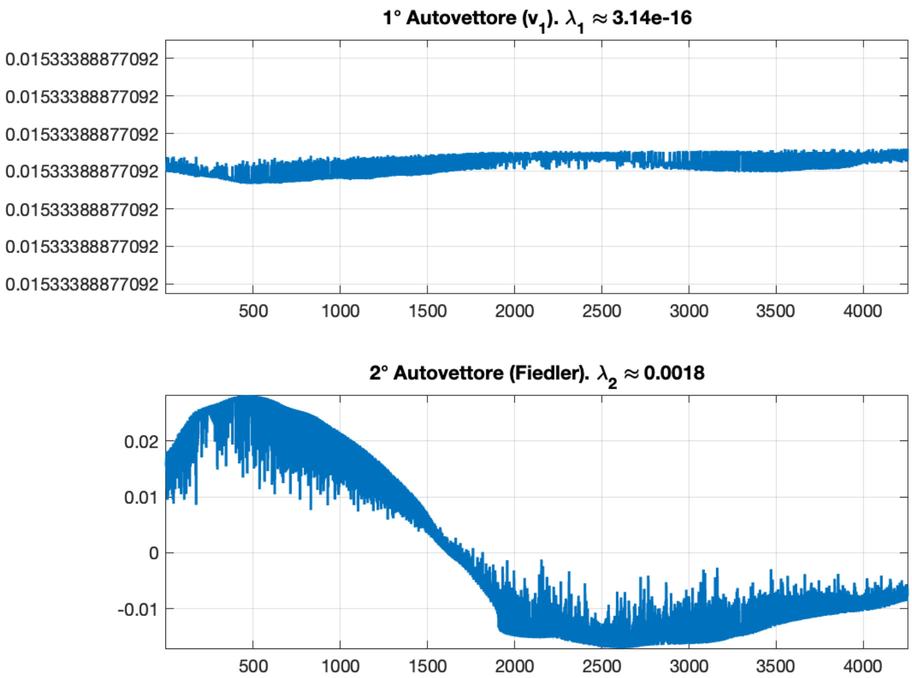


Figure 1: First and Second Eigenvectors of the `airfoil1` graph Laplacian.

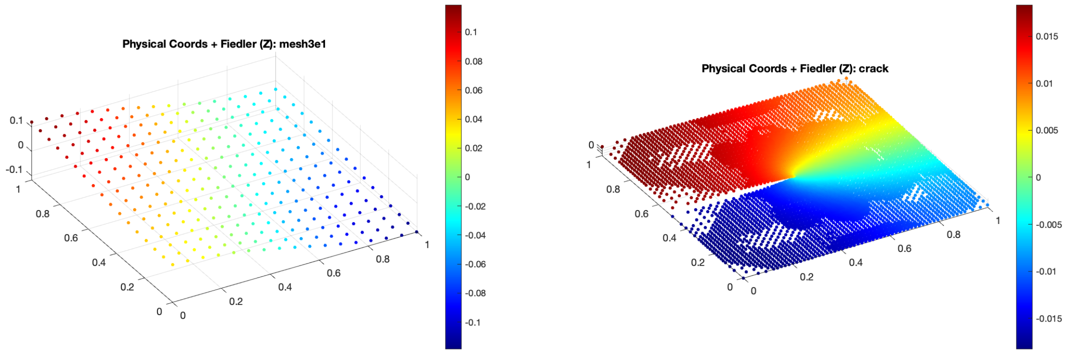
Comment on visual results: This behavior is fully expected.

- The first eigenvector v_1 (top plot) corresponds to the eigenvalue $\lambda_1 \approx 0$ (computed as $3.14e^{-16}$). Since the graph is connected, the multiplicity of the zero eigenvalue is 1, and the associated eigenvector is constant ($v_1 = c \cdot \mathbf{1}$).
- The second eigenvector v_2 (bottom plot) is the *Fiedler Vector*. It is orthogonal to v_1 and its values vary smoothly across the graph indices, crossing zero. This variation provides the

heuristic for spectral bisection: nodes with $v_2 < 0$ belong to one partition, and nodes with $v_2 \geq 0$ to the other.

2.7. Fiedler Vector Projection (Physical Space)

We visualized the values of the Fiedler vector (v_2) by projecting them onto the physical coordinate system of four different meshes. The value of v_2 is represented by both the Z-axis height and the color map.



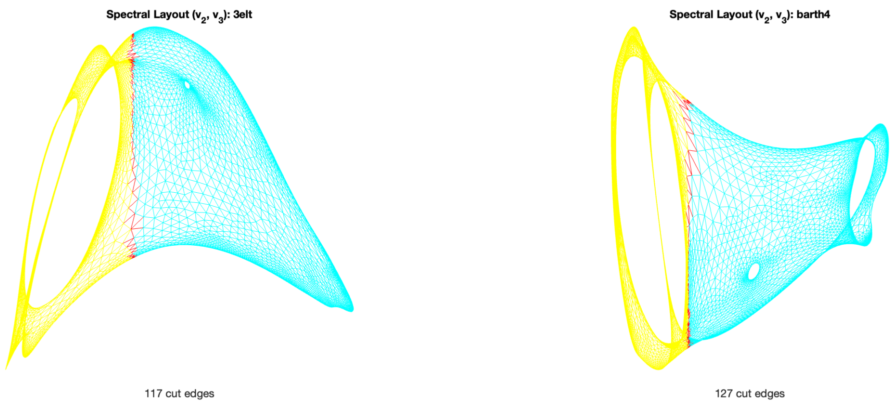
(a) mesh3e1

(b) crack

Figure 2: Projection of the Fiedler vector (v_2) values onto the physical coordinates of the meshes. The color gradient shows how the spectral value varies smoothly across the geometry, identifying the optimal cut.

2.8. Spectral Graph Drawing

Finally, we performed spectral graph drawing by utilizing the second (v_2) and third (v_3) eigenvectors as Cartesian coordinates (x, y) for the nodes. The results for `3elt` and `barth4` are shown below.



(a) 3elt Spectral Layout

(b) barth4 Spectral Layout

Figure 3: Spectral Graph Drawing using coordinates (v_2, v_3). The colors (Cyan/Yellow) represent the bisection cut. In this spectral space, nodes are clustered by connectivity rather than physical distance, making the partition boundary appear as a clear line.

The spectral layout reveals the intrinsic topology of the graphs. Nodes that are highly connected are placed close together in this space, "unfolding" the graph such that the bisection cut appears geometrically simple.

3. Task: Quality of the Report [15 Points]

Additional notes and submission details

Submit the source code files (together with your used `Makefile`) in an archive file (tar, zip, etc.), and summarize your results and the observations for all exercises by writing an extended Latex report. Use the Latex template from the webpage and upload the Latex summary as a PDF to iCorsi.

- Your submission should be a gzipped tar archive, formatted like project_number_lastname_firstname.zip or project_number_lastname_firstname.tgz. It should contain:
  - all the source codes of your MATLAB solutions;
  - your write-up with your name project_number_lastname_firstname.pdf.
- Submit your .zip/.tgz through Icorsi.