ETH zürich

**High-Performance Computing Lab** **2021**

Student: Sina Klampt Discussed with: Alain Hügli

---

# Solution for Project 5

Due date: 17.05.2021 (midnight)

---

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

In the first part we had to install METIS and check whether the interface was working using a small code snippet.

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

To read and create the plots I first read all the csv files in. These were in the csv folder under datasets/Countries/csv/. I then created a matrix for each country with the correct number of nodes respectively and initialized it to 0. Then I looped over each country and set the index to 1 when there existed and edge from i to j. With the sparse() command, I was able to make the matrices sparse. Finally I was able to plot the graphs using the gplotg function which was already given in the exercise. You can see my results in the following plots (Figures 1 through 7):

Figure 1: GB graph

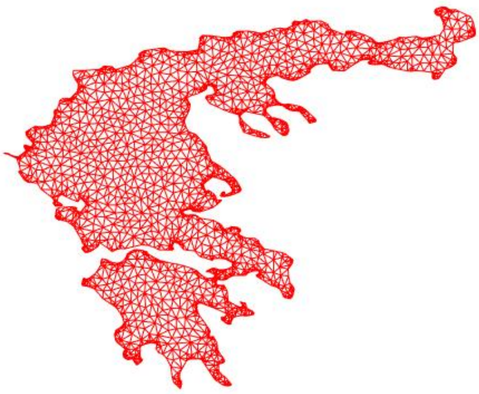

Figure 2: GR graph

Figure 3: CH graph

Figure 4: CL graph

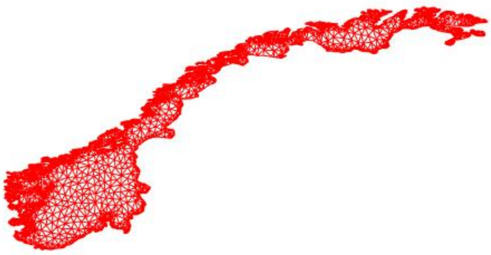

Figure 5: NO graph

Figure 6: RU graph

Figure 7: VN graph

# 3. Task: Implement various graph partitioning algorithms [25 points]

## 3.1. Spectral Graph Bisection

To create the Laplacian matrix, I needed the diagonal matrix D which has the sum of the corresponding row of the matrix A as their entry. So to generate D, I just used the sum() function provided by Matlab. After generating D it was easy to create L. With the eigs() function provided by Matlab I found the eigenvector corresponding to the second smallest eigenvalue and set the threshold to the median. I then set the i-th entry of the map to 1 if the entry of the fiedler vector was smaller than the threshold.

## 3.2. Inertial Graph Bisection

For this subtask I created 2 vectors x and y and calculated the mean for each vector. With that mean I was able to calculate the elements of the matrix M. I then used the eigs() function provided by Matlab again to calculate the eigenvalue decomposition. This time I saved the eigenvector corresponding to the smallest eigenvalue in u. Then divided u by its norm.

You can see my results in the following table (Table 1):

Table 1: Bisection results

| Mesh | Coordinate | Metis 5.0.2 | Spectral | Inertial |
|---|---|---|---|---|
| mesh1e1 | 18 | 17 | 18 | 18 |
| mesh2e1 | 37 | 37 | 35 | 37 |
| netz4504_dual | 25 | 20 | 23 | 26 |
| stufe | 16 | 35 | 16 | 16 |

## 4. Task: Recursively bisecting meshes [15 points]

The implementation in this part was rather straightforward. We needed to call the function 'rec bisection' with the respective methods and choose the parameter for the number of levels correctly. Eventhough the colors in the plots are different they still look all the same which was confusing to me.

You can find my results for all the cases in the following table (Table 2) and the plots for the case "crack" (Figures 8 through 11):

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

| Case | Spectral | Metis 5.0.2 | Coordinate | Inertial |
|---|---|---|---|---|
| mesh3e1 | 121 | 113 | 122 | 184 |
| airfoil1 | 631 | 565 | 819 | 1282 |
| 3elt | 752 | 693 | 1168 | 1342 |
| barth4 | 835 | 713 | 1306 | 1553 |
| crack | 1419 | 1253 | 1860 | 2183 |

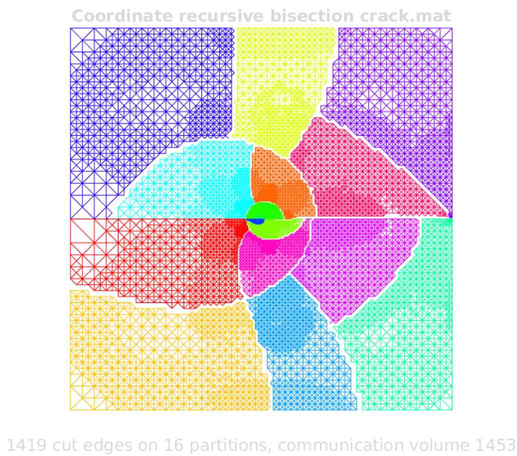


Figure 8: Coordinate bisection

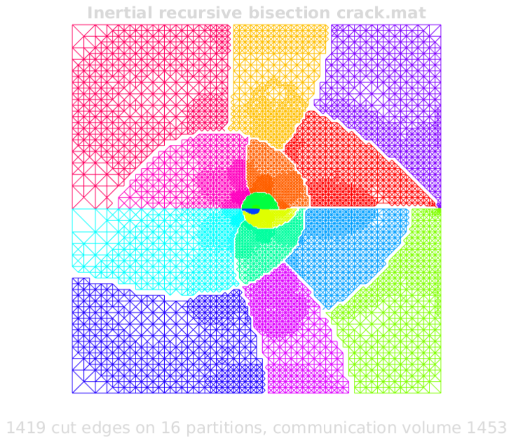


Figure 9: Inertial bisection

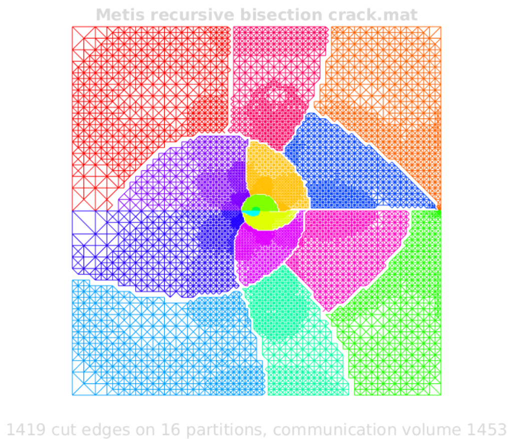


Figure 10: Metis bisection

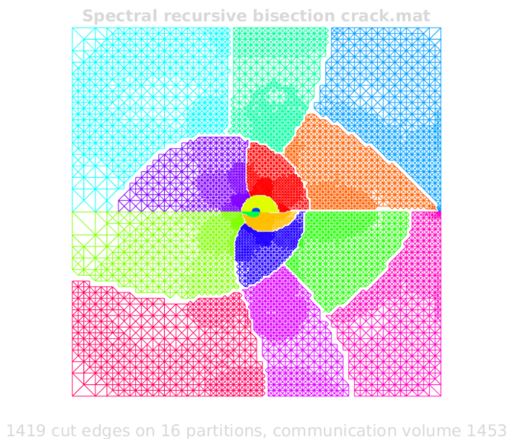


Figure 11: Spectral bisection

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

In this task we had to visualize the road maps for the graphs obtained in subtask 2. I encountered a couple problems when implementing this task. First I could not reproduce the graphs from subtask 2 because my graphs were stored as tuple of matrices whereas the two graphs already implemented in the code, us_roads and luxembourg_osm, were stored in a struct with different information. Another problem was also that I got an error stating:

'Error using checkArgsForHandleToPrint Handle input argument contains nonhandle values.'

Unfortunately I did not have the time to find and correct these errors. And therefore do not have any results to show here.

## 6. Task: Utilizing graph eigenvectors [25 points]

### 6.1. Plot the entries of the eigenvectors associated with the first and second smallest eigenvalues

The values of the second eigenvector are equally distributed around 0 which is exactly what we have expected. You can see my result in the following plot (Figure 12):

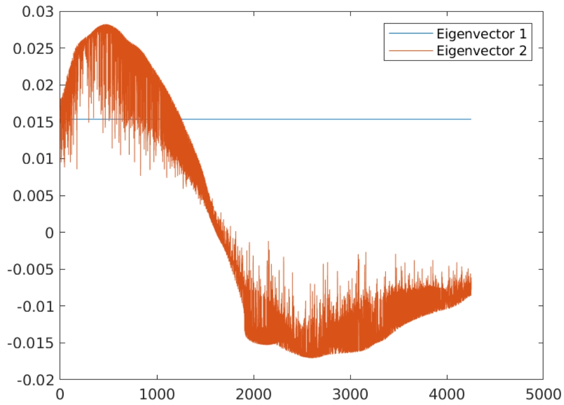


Figure 12: Eigenvector comparison airfoil1

### 6.2. Plot the entries of the eigenvectors associated with the second smallest eigenvalue

To read the relevant parameters, I copied a lot of code from the other files. You can see the eigenvector projection plots in the following plots (Figure 13 through 16):

### 6.3. The spectral bi-partitioning results using the spectral coordinates of each graph

For this subtask you can see my results in the following plots (Figure 17 through 20):

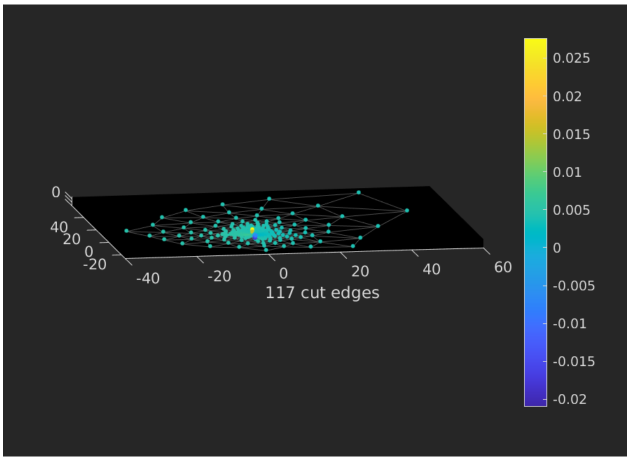


Figure 13: Eigenvector projection on 3elt

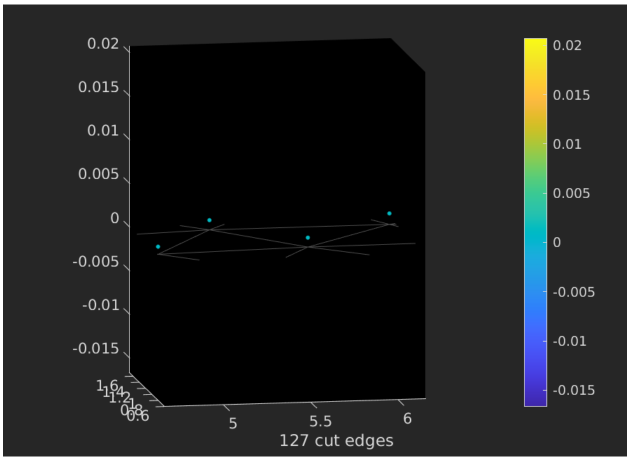


Figure 14: Eigenvector projection on barth4

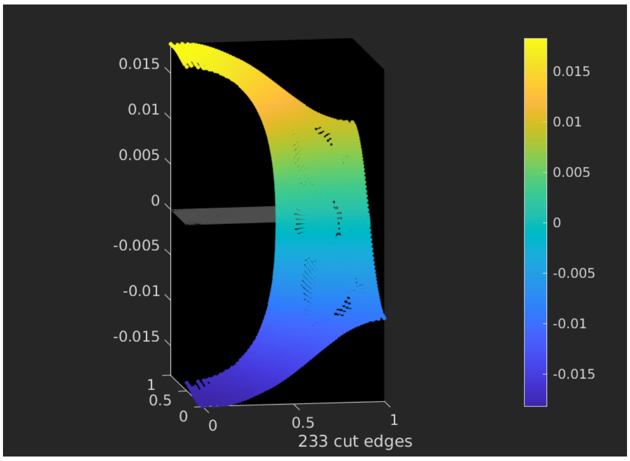


Figure 15: Eigenvector projection on crack

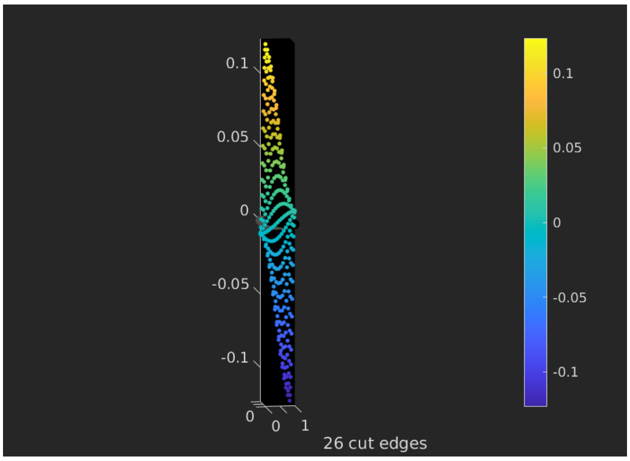


Figure 16: Eigenvector projection on mesh3e1

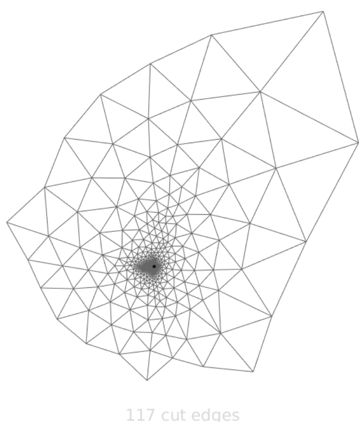


Figure 17: spectral bi-partitioning 3elt

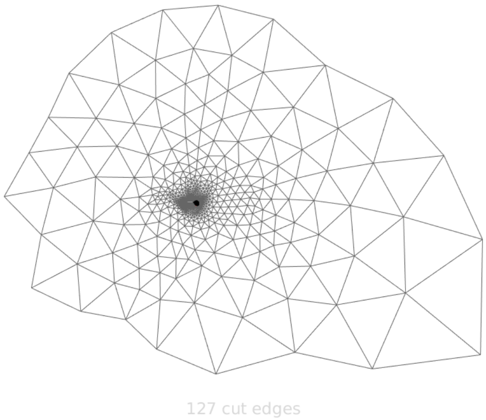


Figure 18: spectral bi-partitioning barth4

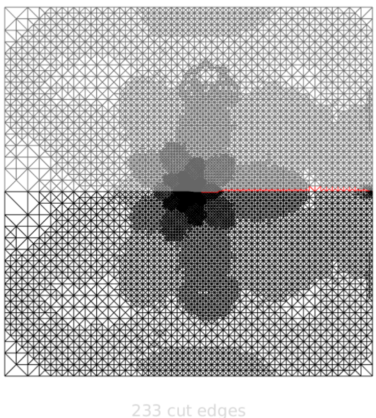


Figure 19: spectral bi-partitioning crack

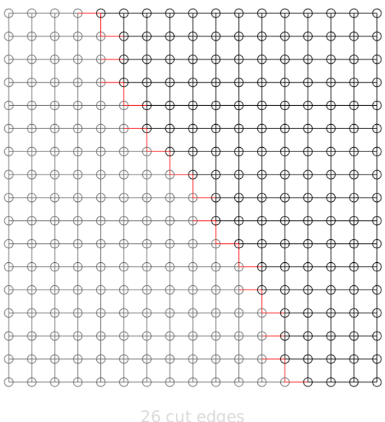


Figure 20: spectral bi-partitioning mesh3e1