Università
della
Svizzera
italiana

**Institute of
Computational
Science
ICS**

Numerical Computing **2020**

Student: Claudio Maggioni Discussed with: –

Solution for Project 3

Due date: Wednesday, 4 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.

Please note all scripts used can be found under the folder `Project_3_Maggioni_Claudio/src`.

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

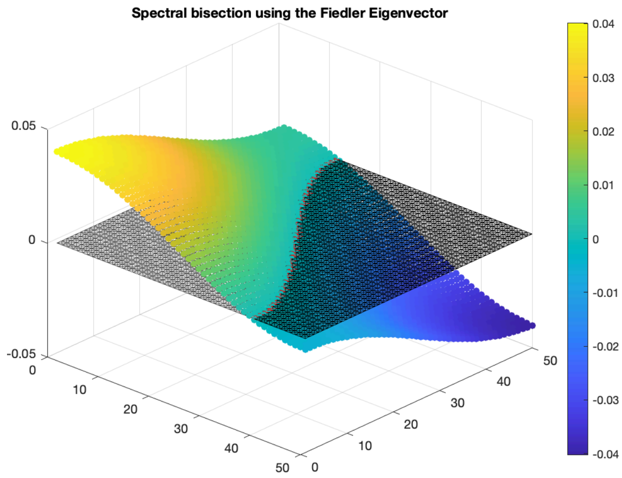
2. Implement various graph partitioning algorithms *(60 Points)*

I summarize the various benchmark results in table 1. Please note that this table can be generated at will with the script `ex2_bisection_table.m`.

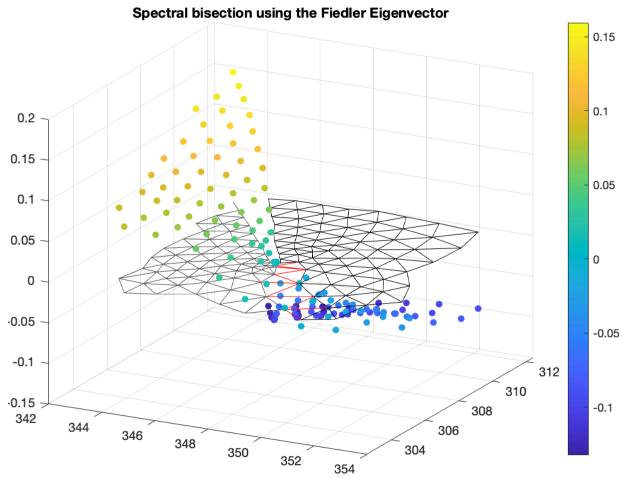
3. Visualize the Fiedler eigenvector *(10 Points)*

In figure 1 there are graph outputs respectively from *Grid9*, *Small*, and *Eppstein*. Please note that these results can be reproduced using script `ex3_plots.m`.

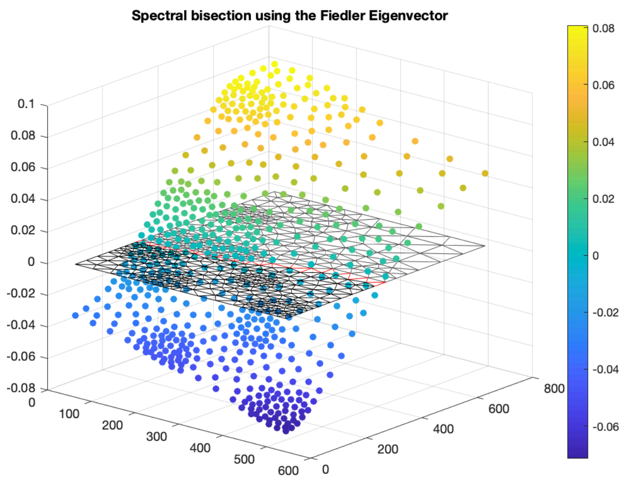
Colors to represent the partitions and the eigenvector components are the same as the ones used in the the assignment's Figure 3. It is quite easy to evince from these plots that spectral partitioning indeed partitions the graph verticies based on the sign of their respective Fiedler eigenvector component: the only intersection between the Graph plane and the surface the eigenvector entries lay corresponds to the edgecut produced by spectral partitioning. This is due to the fact that in the figure the graph is positioned at $z = 0$.



(a) Plot for *Grid9*



(b) Plot for *Small*



(c) Plot for *Eppstein*

Figure 1: Graph outputs for the 3 adjacency matrices.

4. Recursively bisecting meshes *(20 Points)*

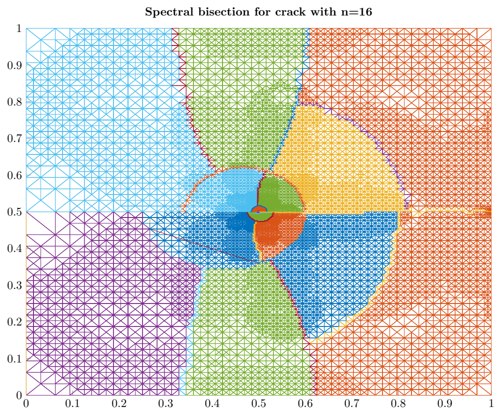
I summarize my results in table 2. Additionally, the graph plots for a recursive partition in 16 parts of *Crack* are avaliable in figure 2.

Again the data shown here can reproduced using a MATLAB script (`Bench_rec_bisection.m`). Please note that the given recursion helper function `rec_bisection` was not used. Instead, the function `runtest` was implemented to allow for partitioning graphs to be generated at will.

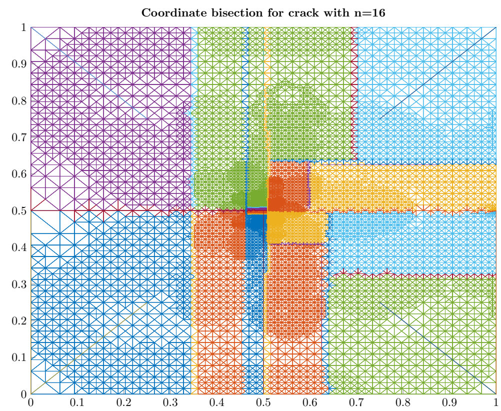
5. Compare recursive bisection to direct k-way partitioning *(10 Points)*

I summarize my results in table 3. Additionally, plots for the $n = 32$ cases can be found in figure 3. Please note that both the table and the plots can be reproduced using script `Bench_metis.m`.

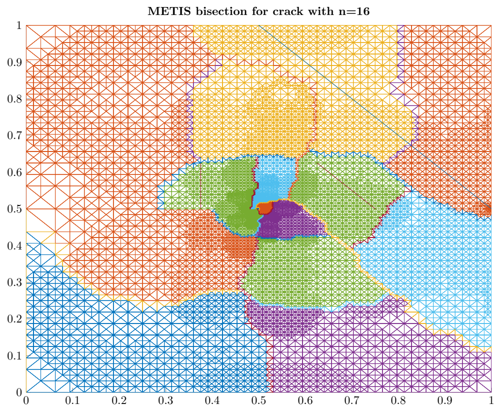
I had the suspicion that k-way partitioning would perform better in all cases, since in this the partitioning algorithm has a chance to see the entire problem and to perform decision that could affect all partitions. Instead, for the `Crack` mesh, recursive partitioning surprisedly performed better than k-way for both $n = 16$ and $n = 32$, producing slighly smaller edge cuts.



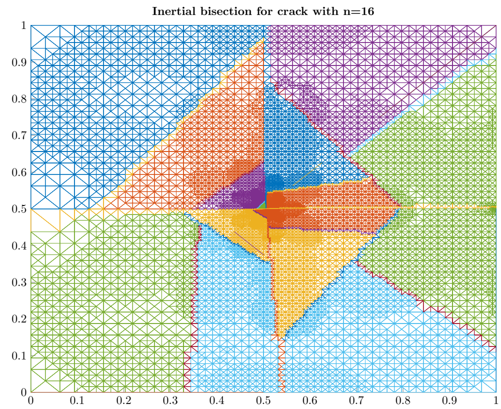
(a) Spectral bisection



(b) Coordinate bisection



(c) METIS bisection



(d) Inertial bisection

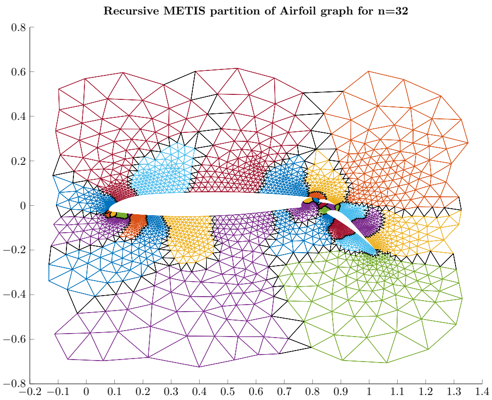
Figure 2: Graph outputs for *Crack* graph with $n = 16$

Table 1: Bisection results

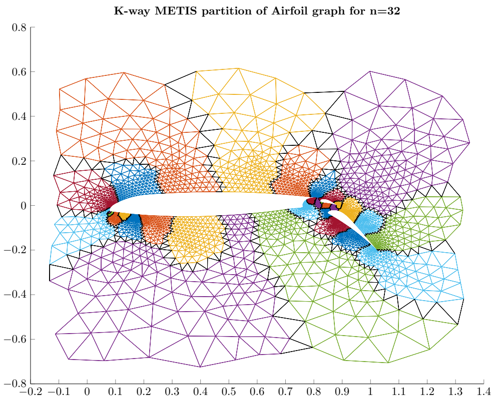
Mesh	Coordinate	Metis 5.0.2	Spectral	Inertial
grid5rect(10,100)	10	10	10	10
grid5rect(100,10)	10	10	10	10
grid5recRotate(100,10,-45)	18	10	10	10
gridt(40)	58	58	58	58
grid9(30)	88	92	104	88
Smallmesh	25	13	12	30
Tapir	55	34	18	49
Eppstein	42	48	45	45

Table 2: Edge-cut results for recursive bi-partitioning (data for $n = 8$ on the left and $n = 16$ on the right).

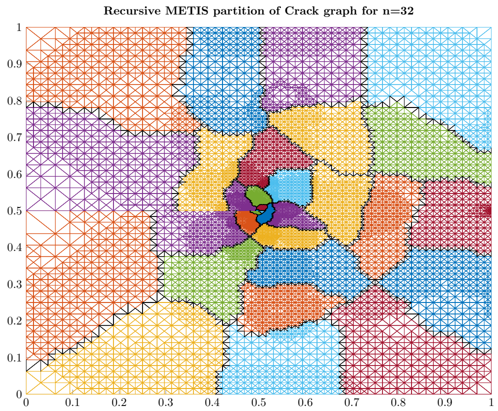
Case	Spectral		Metis 5.0.2		Coordinate		Inertial	
airfoil1	327	578	320	563	516	819	577	897
3elt	372	671	395	651	733	1168	880	1342
barth4	505	758	405	689	875	1306	891	1350
mesh3e1	72	111	75	117	75	122	67	102
crack	804	1303	784	1290	1343	1860	1061	1618



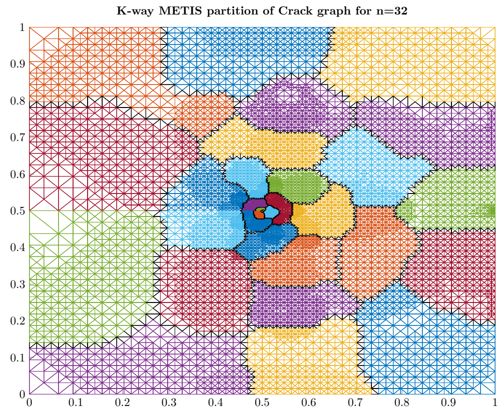
(a) *Airfoil* with recursive partitioning



(b) *Airfoil* with k-way partitioning



(c) *Crack* with recursive partitioning



(d) *Crack* with k-way partitioning

Figure 3: Graph outputs for *METIS* recursive and k-way partitioning with $n = 32$. Edge cuts are displayed in black

Table 3: Comparing the number of cut edges for recursive bisection and direct multiway partitioning in Metis 5.0.2. Results for recursive partitioning shown on the left, k-way on the right.

Partitions	crack	airfoil1
16	10055 10077	580 564
32	10824 10943	967 947