Università della Svizzera italiana | Faculty of Informatics | Institute of Computing CI

Numerical Computing **2022**

Student: Kelvin Likollari Discussed with: Gerald Prendi

Solution for Project 3

Due date: Wednesday, November 9, 2022, 11:59 PM

Numerical Computing 2022 — 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, Julia). 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. The assignment

1.1. Implement various graph partitioning algorithms [50 points]

The following section contains the bisection edge-cut results for all toy meshes. The results are presented in tabular view.

Table 1: Bisection results

Mesh	Coordinate	Metis 5.0.2	Spectral	Inertial
grid(12, 100)	12	12	12	14
grid(100, 12)	12	12	12	14
grid(100, 12, pi/4)	22	14	12	14
gridt(50)	73	80	66	109
gridt(40)	59	62	54	86
Smallmesh	25	13	12	16
Tapir	55	24	18	79
Eppstein	42	40	42	65

The threshold value chosen for the spectral graph bisection is 0. The reason why this particular threshold was chosen among others is that it is firstly more efficient. Specifically, having a threshold value of 0 performs more efficiently compared to having a median threshold value. As the given PDF states, having a threshold equal to 0 will yield a roughly balanced partition with minimum edge-cut. Choosing the median value as a threshold produces strictly balanced partitions and is less efficient than choosing the zero value as a median.

A comment can be derived based on the results of table 1. Effectively, the bisection edge-cut values for every one of the meshes, range in a relatively small interval of numbers (i.e. [12, 120]). Additionally, the results for the first 2 meshes are the same for the first 3 edge-cut types, whereas, in the inertial bisection, the values for the first three meshes are the same. After moving from grid meshes to the other kind of meshes (i.e Smallmesh, Tapir, Eppstein), the difference between the different edge-cuts tends to increase, compared to the different edge-cut applies to the grid meshes.

1.2. Recursively bisecting meshes [20 points]

Table 2: Edge-cut results for 3-level recursive bi-partitioning.

Case	Spectral	Metis 5.1.0	Coordinate	Inertial
mesh3e1	59	68	75	90
airfoil1	320	410	524	590
3elt	550	618	720	850
barth4	415	555	695	600
crack	757	960	1014	1050

Table 3: Edge-cut results for 4-level recursive bi-partitioning.

Case	Spectral	Metis 5.1.0	Coordinate	Inertial
mesh3e1	112	90	125	180
airfoil1	662	573	865	950
3elt	759	1012	1260	1660
barth4	860	750	1220	1117
crack	1515	1434	1955	1380

The results displayed in tables 2 and 3 are the values derived for every mesh edge-cut on a 3 and 4-level recursive bi-partition. An observation that can be made in this scenario is that the greater the edge cut is, the bigger the consequence. A brief glance at the two tables above, reveals that the values of 4-level recursive bi-partitioning are roughly double compared to those of 3-level recursive bi-partitioning. This doubling pattern shall be followed also in the scenarios when employing a 5-level cut, a 6-level cut, etc. Finally, the conclusion derived is that the bigger the cut level, the greater the future edge-cut results will be.

1.3. Comparing recursive bisection to direct k-way partitioning [15 points]

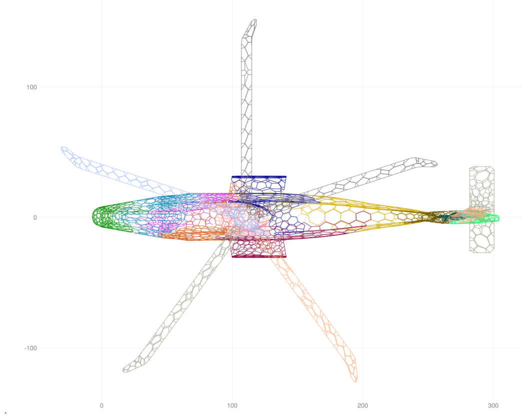
The following section contains, in a tabular form, the number of cut edges for recursive bisection and direct multi-way partitioning.

Table 4: Cut edges for recursive bisection and direct multi-way partitioning in Metis 5.1.0.

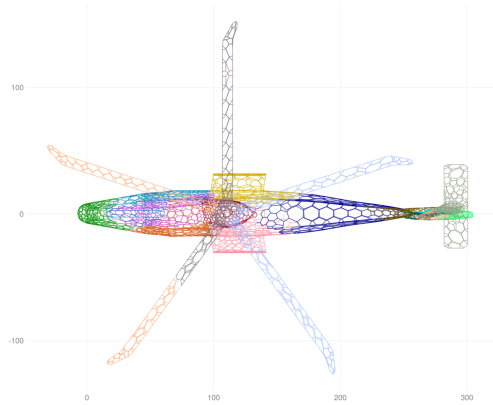
Partitions	Helicopter	Skirt
16-recursive bisection	377	3262
16-way direct bisection	336	3307
32-recursive bisection	578	6364
32-way direct bisection	528	6095

A comment that can be made based on the results derived on table 4 is that the use of recursive graph partitioning is significantly less efficient than the direct k-way partitioning, since as the provided PDF states, recursive bisection is dependent in a great extent on the decisions made during the early stages of the process, and also suffers from the lack of global information. Henceforth, it may result in sub-optimal partitions and this yields the conclusion that direct k-way partitioning can be acquired to improve the partitions performed by the recursive graph partitioning.

The following section depicts the visualization for the different partitions, starting with the helicopter visualization for all the respective cases:

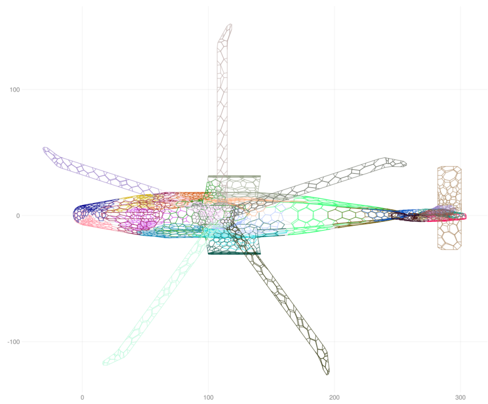


(a) 16 Recursive Helicopter

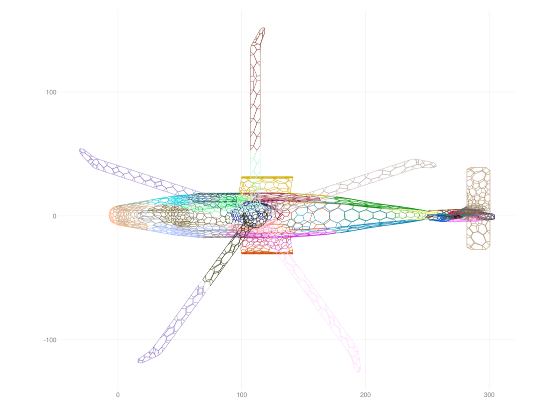


(b) 16 K-Way Helicopter

Figure 1: 16 - Helicopter Direction



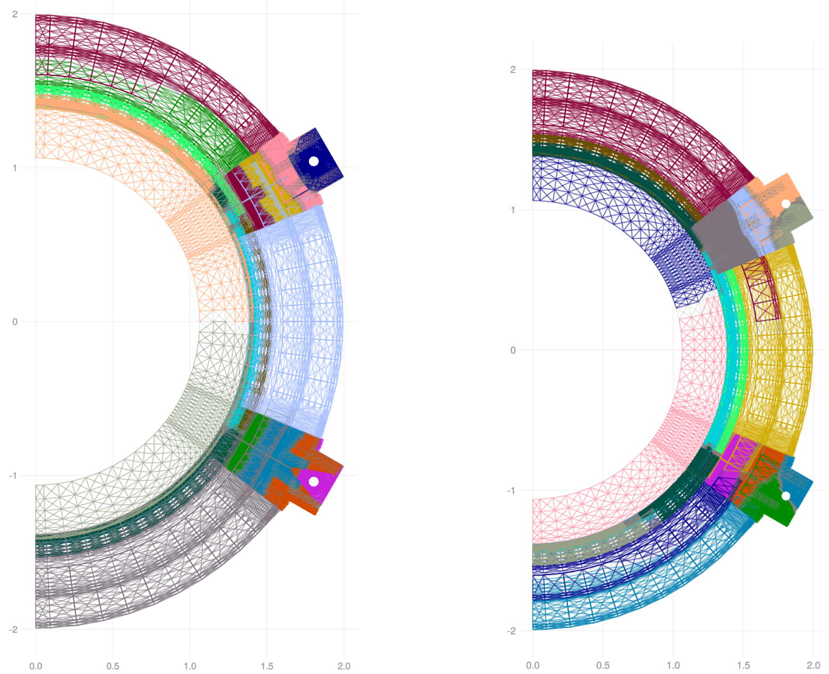
(a) 32 Recursive Helicopter



(b) 32 K-Way Helicopter

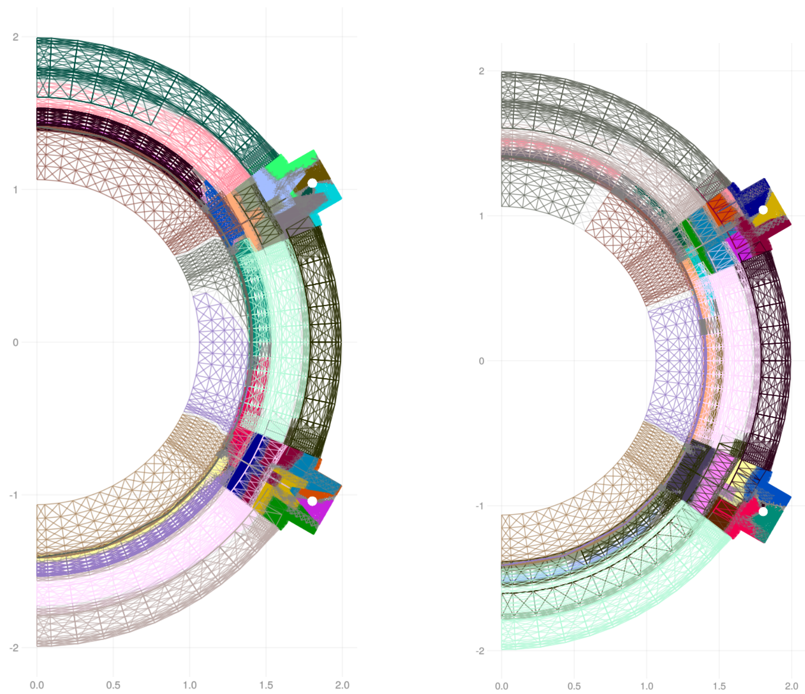
Figure 2: 32 - Helicopter Direction

The following section contains the visualization for the skirt:



(a) 16 Recursive Skirt (b) 16 K-Way Skirt

Figure 3: 16 - Skirt Direction



(a) 32 Recursive Skirt (b) 32 K-Way Skirt

Figure 4: 32 - Skirt Direction

1.4. References

1. Berkley EECS