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

Numerical Computing **2023**

Student: Birindelli Leonardo

Solution for Project 2 **Due date:** Wednesday, 25 October 2023, 11:59 PM

---

1. The assignment

1.1. Implement various graph partitioning algorithms [50 points]

1.1.1. Implementation of the spectral bisection algorithm

First of all, I create the Laplacian matrix from the sparse matrix A :

```

% 1. Construct the Laplacian matrix.
D = diag(sum(A));
L = D - A;
```

Secondly, I aim to calculate the eigendecomposition of that matrix:

```

% 2. Calculate its eigensdecomposition

% returns the 2 smallest eigevalues , and their corrispective eigevectors , of
    the Laplacian matrix L
 [V,D]=eigs(L,2,1e-6);
 feidler_vector=V(:,2);
```

In conclusion, once I obtained the Fiedler vector and calculate the median of all components of that vector, I partition the vertices of the graph into the two section of the structure

```
% 3. Label the vertices with the components of the Fiedler vector.
 feidler_median=median(feidler_vector,"all"); %threshold

% 4. Partition them around their median value, or 0.
 part1=find(feidler_vector<feidler_median);
 part2=find(feidler_vector>=feidler_median);
```

1.1.2. Implementation of the inertial bisection algorithm

First of all, I calculate the center of mass where its coordinates are computed in the following way:

$x = \frac{1}{n}\sum_{i=1}^{n} x_i$

$y = \frac{1}{n}\sum_{i=1}^{n} y_i$

```
% 1. Calculate the center of mass.
%Coordinates are the following
x_centerMass = sum(xy(:,1)) / size(xy, 1);
y_centerMass = sum(xy(:,2)) / size(xy, 1);
```

Secondly, I aim to calculate the symmetric matrix M after calculating the sums $S_{\mathrm{xx}}, S_{\mathrm{xy}}, S_{\mathrm{yy}}$ in which :

$S_{\mathrm{xx}} = \sum_{i=1}^{n}(x_i - x)^2$

$S_{\mathrm{xy}} = \sum_{i=1}^{n}(x_i - x)(y_i - y)$

$S_{\mathrm{yy}} = \sum_{i=1}^{n}(y_i - y)^2$

```
% 2. Construct the matrix M.
M = zeros(2, 2);

%Calculate S_xx
M(1,1) = sum(xy(:,1)-x_centerMass)^2 ;

%Calculate S_xy
y_diff=xy(:,2)-y_centerMass;
x_diff=xy(:,1)-x_centerMass;
M(1,2) = sum(x_diff.*y_diff) ;
M(2,1) = M(1,2);

%Calculate S_yy
M(2,2) = sum(xy(:,2)-y_centerMass)^2 ;
```

Thirdly, I compute the smallest eigenvector of the matrix M:

```
% 3. Calculate the smallest eigenvector of M.

% returns the smallest eigevalues, and its corrispective eigevector, of the matrix M
[V,~]=eigs(M,1,1e-6);
u_vector=V(:,1);

%gets 2 first values of smallest eigenvector
u_vector=u_vector(1:2);
a=u_vector(1); %u1
b=u_vector(2); %u2
```

In conclusion, I use the function "partition.m" to compute the partition in two sections of the graph in order to complete the partitioning operation

```
% 4. Find the line L on which the center of mass lies.
% 5. Partition the points around the line M.
[part1,part2] = partition(xy,[a,b]);
```

The next tables show bisection results obtained by applying to each mesh different bisection algorithms

Table 1: Bisection results

Mesh	Coordinate	Metis 5.0.2	Spectral	Inertial
grid5rec(12,100)	12	12	12	12
grid5rec(100,12)	12	12	12	100
grid5recRotate(100,12,-45)	22	12	12	100
gridt(50)	72	82	80	82
grid9(40)	118	127	120	118
Smallmesh	25	12	14	16
Tapir	55	23	58	81
Eppstein	42	41	47	65

1.2. Recursively bisecting meshes [20 points]

The subsequent tables show bisection results obtained by applying to each mesh different bisection algorithms. The former board shows the results of 8-way partitions and the latter one lists the computations of 16-way partitions :

Table 2: Edge-cut results for 8-way partitioning.

Case	Spectral	Metis 5.1.0	Coordinate	Inertial
mesh3e1	58	57	63	78
bodyy4	1195	985	1065	2224
de-2010	2818	491	929	1745
biplane-9	510	465	548	1435
L-9	704	637	631	1963

Table 3: Edge-cut results for 16-way partitioning.

Case (8 , 16)	Spectral	Metis 5.1.0	Coordinate	Inertial
mesh3e1	58	57	63	78
bodyy4	1959	1591	1951	4226
de-2010	5178	897	1796	3665
biplane-9	895	845	974	2931
L-9	1119	1019	1028	3951

The next images shows the results obtained for the case named "de-2010" in the application of the 8-way and 16-way partitions

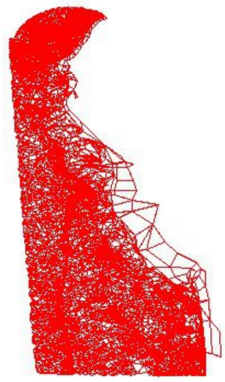

Figure 1: "de-2010" original graph

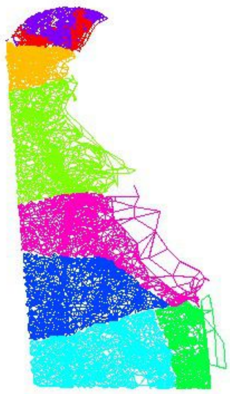

Figure 2: 8-way "de-2010" graph

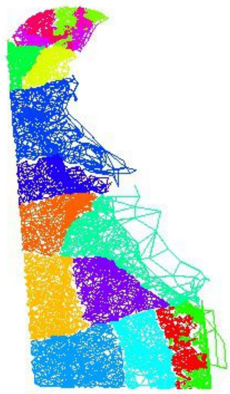

Figure 3: 16-way "de-2010" graph

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

1. Which one did you expect to perform better and why?

   Despite the fact that both algorithms are excellent and their efficiency depends directly on the problem to which they are applied, I generally expected the best algorithm between the two methods to be that of k-way partitioning because, unlike the recursive bisection method where the final result is strictly dependent on the choices made during the initial stages of the partitioning process, the graph is divided into k balanced partitions without the aid of recursive partitioning as is the case with the first algorithm. Therefore the final result is derived without any influence from the decisions made in the initial stages.

2. Was your guess confirmed by the results of the example meshes?

   No , it was not . Because as I said before the efficiency of both methods strictly depends on the problem. Indeed,for example, the 32-way "helicopter.m" case results to has a better number of cut edges when the recursive bisection algorithm is applied than the other one.

Table 4: Edge-cut results for the two algorithms

Partitions	Helicopter	Skirt
16 - recursive bisection	343	3119
16-way direct partition	324	3393
32 - recursive bisection	537	6075
32-way direct partition	539	6051

Visualization of the partitioning results for both graphs for 32 partitions.

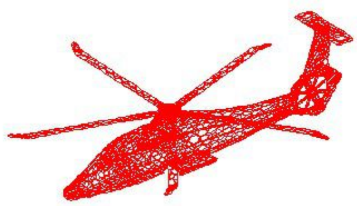

Figure 4: "helicopter.m" graph

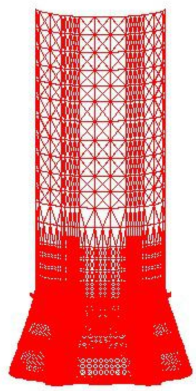

Figure 5: "skirt.m" graph

The following images show the application of the "32 - recursive bisection" to each case

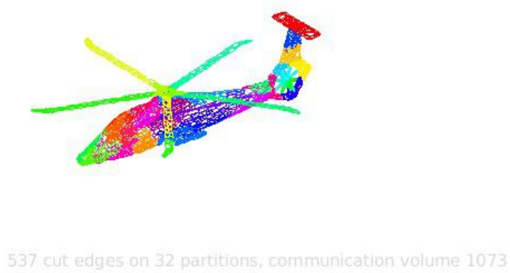

Figure 6: "helicopter.m" graph

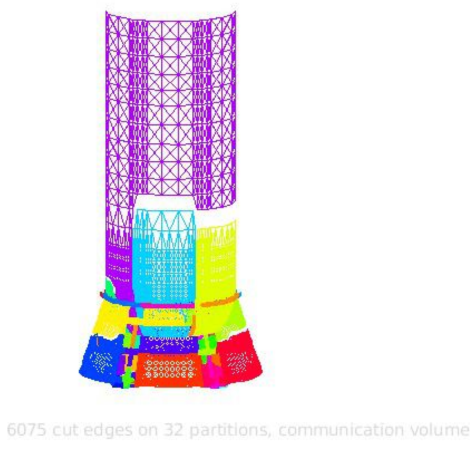

Figure 7:

On the other hand, these pictures show the application of the "32 - way partition" to each case

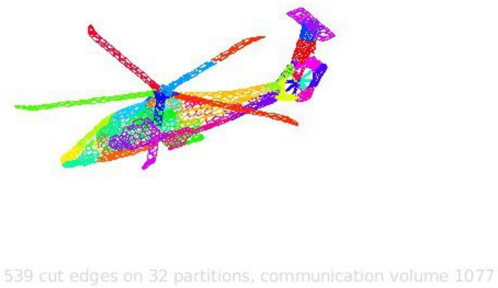

Figure 8: "helicopter.m" graph

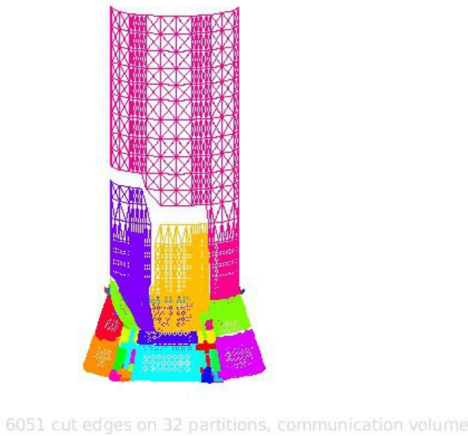

Figure 9:

1.4. References

1 "Introduction to graph partitioning" slides of the lesson of 4th October 2023 avaiable on iCorsi platform

2 Aydin Buluc, Henning Meyerhenke, Ilya Safro, Peter Sanders, and Christian Schulz. Recent Advances in Graph Partitioning, pages 117–158. Springer International Publishing, Cham, 2016

3 Timothy A. Davis and Yifan Hu. The University of Florida Sparse Matrix Collection. *ACM Transactions on Mathematical Software*, 38(1):1:1–1:25, December 2011.

4 U. Elsner. Graph Partitioning: A Survey. Technical report, Technische Universität Chemnitz, Germany, 97-27, 1997.

5 C. M. Fiduccia and R. M. Mattheyses. A Linear-Time Heuristic for Improving Network Partitions. In *Proceedings of the 19th Design Automation Conference, DAC '82*, pages 175–181, Piscataway, NJ, USA, June 1982. IEEE Press.

6 George Karypis and Vipin Kumar. Parallel Multilevel K-Way Partitioning Scheme for Irregular Graphs. In *Proceedings of the 1996 ACM/IEEE Conference on Supercomputing, Supercomputing '96*, page 35–es, USA, 1996. IEEE Computer Society.

7 B. W. Kernighan and S. Lin. An Efficient Heuristic Procedure for Partitioning Graphs. *The Bell System Technical Journal*, 49(2):291–307, February 1970.

8 Horst D. Simon and Shang-Hua Teng. How Good Is Recursive Bisection? *SIAM Journal on Scientific Computing*, 18(5):1436–1445, September 1997.