Università
della
Svizzera
italiana

Institute of
Computational
Science
ICS

High Performance Computing **2018**

Instructor: Prof. Olaf Schenk TA: Juraj Kardos, Radim Janalik

Assignment 7

Due date: Monday, 10 December 2018, 23:55

In high-performance computing, the graph partition problem is defined on data represented in the form of a graph $G = (V, E)$, with V vertices and E edges, such that it is possible to partition G into smaller components with specific properties. For instance, a k-way partition divides the vertex set into k smaller components. A good partition is defined as one in which the number of edges running between separated components is small. Uniform graph partition is a type of graph partitioning problem that consists of dividing a graph into components, such that the components are of about the same size and there are few connections between the components. Important applications of graph partitioning include scientific computing, partitioning various stages of a VLSI design circuit and task scheduling in multi-processor systems.

Recently, the graph partition problem has gained importance due to its application for clustering and detection of cliques in social, pathological and biological networks. A survey on the recent trends in computational methods and applications can be found in [1]. Since graph partitioning is a hard problem, practical solutions are based on heuristics. There are two broad categories of methods, local and global. Well known local methods are the Kernighan–Lin algorithm, and Fiduccia-Mattheyses algorithms, which were the first effective 2-way cuts by local search strategies. Their major drawback is the arbitrary initial partitioning of the vertex set, which can affect the final solution quality. Global approaches rely on properties of the entire graph and do not rely on an arbitrary initial partition. The most common example is spectral partitioning, where a partition is derived from the spectrum of the adjacency matrix.

The purpose of this assignment is

1. to implement various graph partitioning algorithms in Matlab and to test these methods on a variety of smaller 2D meshes,
2. to use standard software package for graph partitioning (such as METIS) and to use modern computational science software engineering tools such as GitHub and mex to interface external software libraries with Matlab,

3. to evaluate the quality of the multi-level Fiduccia-Mattheyses algorithm, and a spectral partitioning for a 2D finite element mesh from airfoil simulations.

4. to partition realistic finite-element meshes using the METIS graph partitioning code and VISIT for the visualization.

Software tools

We will use two partioning software tools for this HPC miniproject. The first one is METIS[1] [2], which is a graph partitioning family by Karypis and Kumar and it is probably the most well known graph partitioning software. Among this family, kMetis aims at greater partitioning speed, hMetis, applies to hypergraphs and aims at partition quality, and ParMetis is a parallel implementation of the Metis graph partitioning algorithm. The second one is KaHIP (Karlsruhe High Quality Partitioning)[2] [1], which is a multilevel hypergraph partitioning framework providing direct k-way and recursive bisection based partitioning algorithms that compute solutions of very high quality. For the visualization we will use Visit[3] which is an Open Source, interactive, scalable, visualization, animation and analysis tool. From Unix, Windows or Mac workstations, users can interactive visualize and analyze data ranging in scale from small ($<$10 cores) desktop-sized projects to large ($> 10^5$ cores) leadership-class computing facility simulation machines.

METIS is a set of serial programs for partitioning graphs, partitioning finite element meshes, and producing fill reducing orderings for sparse matrices. The algorithms implemented in METIS are based on the multilevel recursive-bisection, multilevel k-way, and multi-constraint partitioning schemes. METIS's key features are the following:

- Provides high quality partitions: Experiments on a large number of graphs arising in various domains including finite element methods, linear programming, VLSI, and transportation show that METIS produces partitions that are consistently better than those produced by other widely used algorithms. The partitions produced by METIS are consistently 10% to 50% better than those produced by spectral partitioning algorithms.

- It is extremely fast: Experiments on a wide range of graphs has shown that METIS is one to two orders of magnitude faster than other widely used partitioning algorithms. Graphs with several millions of vertices can be partitioned in 256 parts in a few seconds on current generation workstations and laptops.

[1] http://glaros.dtc.umn.edu/gkhome/metis/metis/overview
[2] http://algo2.iti.kit.edu/kahip/
[3] https://wci.llnl.gov/simulation/computer-codes/visit

The assignment

1. **Install METIS 5.0.2, KaHIP 2.0, and the corresponding Matlab mex interface [10 points]:**

   In order to use METIS you need the corresponding Matlab interface, since METIS is written in C language. You can use precompiled Matlab interface (`metismex.mexmaci64`) for Mac. Then, all you need to do is to tell Matlab where to find the binary (see `addpath` command in Matlab). You can use the precompiled binary form GitHub repository or iCorsi.

   If you want to compile the package yourself (e.g. if you are using different OS), read carefully the instruction on https://github.com/dgleich/metismex on how to install METIS 5.0.2, how to use `cmake`, `mex`, and Matlab to build a mex interface between MATLAB and METIS. If you do not have Matlab installed on your laptop you might download it from the intranet of our department.

   Copy your Matlab interface "metismex.mexa64" in the `meshpart` directory so that it can be used for the partitioning of meshes using METIS. Alternatively, specify the path where the interface is located (use `addpath` command). Check that the interface works by using the following code snippet:

```
>> A = blkdiag(ones(5),ones(5));
>> A(1,10) = 1; A(10,1) = 1; A(5,6) = 1; A(6,5) = 1;
>> [p1,p2] = metispart(sparse(A))
p1 =
     1     2     3     4     5
p2 =
     6     7     8     9    10
```

   Install and test the partitioning software KaHIP on the same example as well.

2. **Implement various graph partitioning algorithms in Matlab [40 points]:**

   In the GitHub repository, there is the mesh partitioning toolbox in the directory `meshpart`. You can also download `meshpart.tgz` from the iCorsi webpage. This toolbox contains Matlab code for several graph and mesh partitioning methods, e.g., coordinate bisection. It also has routines to generate recursive multiway partitions, vertex separators, and nested dissection orderings; and it has some sample meshes and mesh generators.

   - Run in Matlab the demo program "demo.m" and familiarize yourself with the Matlab codes in the directory `meshpart`.
   - Implement **spectral partitioning** based on the Fiedler eigenvector. Use the incomplete Matlab file `USIspecpart.m` for your solution.
   - Use the **inertial bisection** partitioning code implemented during the class. Use the incomplete Matlab file `USIinertpart.m` for your solution.
   - Report the bisection edge cut for all partitioning methods and the nine meshes that are used in the demo program. Please use Table 1 to report these results. An example of partitioning results for the Airfoil mesh is shown in Figure 1 to Figure 3.

Table 1. Edge-cut Results

Mesh	Coordinate	Metis 5.0.2	KaHIP	Spectral	Inertial
grid5rec(8, 80)	8				
grid5rec(80, 8)	8				
gridt(20)	28				
grid9(30)	88				
small	25				
Tapir	55				
Eppstein	42				
Airfoil	94				
cockroach(60)	2				

3. **Implement in Matlab the recursive k-way partitioning [10 points]:** Report the edge-cut for k-way partitioning for the NASA mesh airfoil, which is available in meshes.dat. Use and/or modify the Matlab file `dice.m`

```
map = dice('USIspecpart',nlevels,Airfoil,Airfoilxy);
```

where e.g. 'USIspecpart' is the particular bisection method that you would like to test (in this case it is spectral bisection). Please use Table 2 to report these results. Visualize the results (graphs) for a partitioning with 16 and 32 subgraphs. An example for METIS 5.0 is shown in Figure 4.

Table 2. Edge-cut results for k-way partitioning and the airfoil mesh.

Mesh	Coordinate	Metis 5.0.2	KaHIP	Spectral	Inertial
k=2					
k=4					
k=8					
k=16					
k=32					

4. **Visualize the graph partitioning [10 points]:** Create a 64×64 2D mesh using the `grid5rec(nx,ny)` function. This is the structure of the 5-point stencil, which was used in the miniapp (assignments 4 and 6). Partition the mesh with Metis graph partitioning algorithm and compare it with hypergraph KaHIP results. Compare edge-cut (i.e. total communication volume) and include the visualization of both partitioning techniques results in the Latex report for 8 partitions, similar to the figure 5. How does it compare to the partitioning we used in the miniapp in terms of communication volume and load balancing?

5. **Partitioning of realistic large-scale FEM meshes [30 points]:** Use the Matlab template

code from iCorsi (object `Mesh`) that reads a given finite-element mesh from a file and preprocesses the data. Do the following tasks:

- use the METIS and KaHIP functionality to partition the mesh into k subgraphs e.g. $k = 4$, $k = 16$ or $k = 32$,
- visualize the partitions with Visit of Paraview add these results to your Latex report,
- compare the Metis with multilevel method from the "KaHIP_NE" and report partitioning time, edgecut and partition balance for both packages.

Use one of the three meshes `747heat`, `BMW2`, or `brain` to accomplish the tasks above (see figure 6). Use table 3 to report the results.

Table 3. Results for 2-way partitioning of the selected FEM mesh.

Metric	Metis 5.0.2	KaHIP
Time (s)		
Partition 1		
Partition 2		
Edge cut		

Submission: Submit the source code files in an archive file (tar, zip, etc) and show the TA the results. Furthermore, summarize your results and the observations for all exercises by writing an extended Latex summary. Use the Latex template from the webpage and upload the extended Latex summary as a PDF to iCorsi.

References

[1] A. Buluç, H. Meyerhenke, I. Safro, P. Sanders, and C. Schulz, *Recent Advances in Graph Partitioning*, Springer International Publishing, Cham, 2016, pp. 117–158.

[2] G. Karypis and V. Kumar, *A fast and high quality multilevel scheme for partitioning irregular graphs*, SIAM Journal on Scientific Computing, 20 (1998), pp. 359–392.

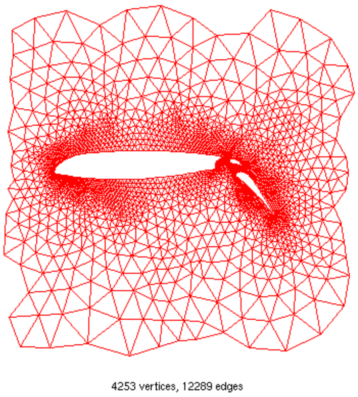
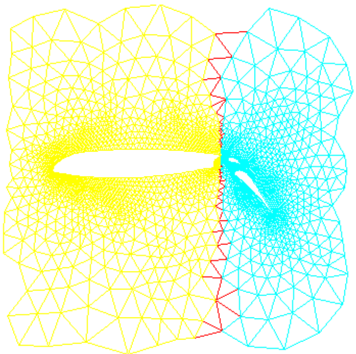


Figure 1. The NASA airfoil that has been designed for supersonic flows.



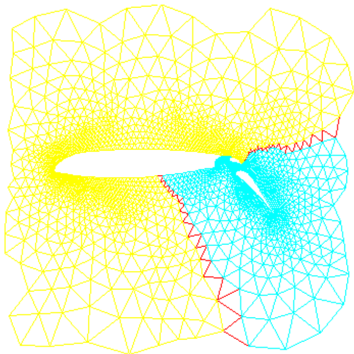
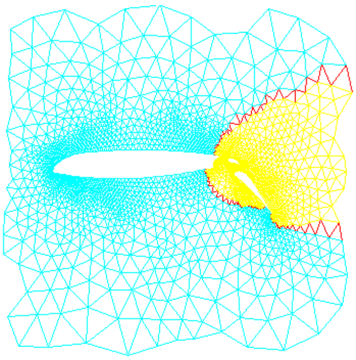


Figure 2. Partitioning results using coordinate bisection (left) and bisection based on Metis 5.0.2 (right).



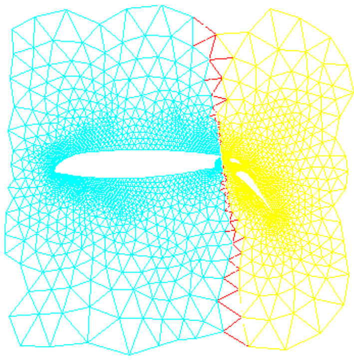


Figure 3. Partitioning results using spectral bisection (left) and inertial bisection (right).

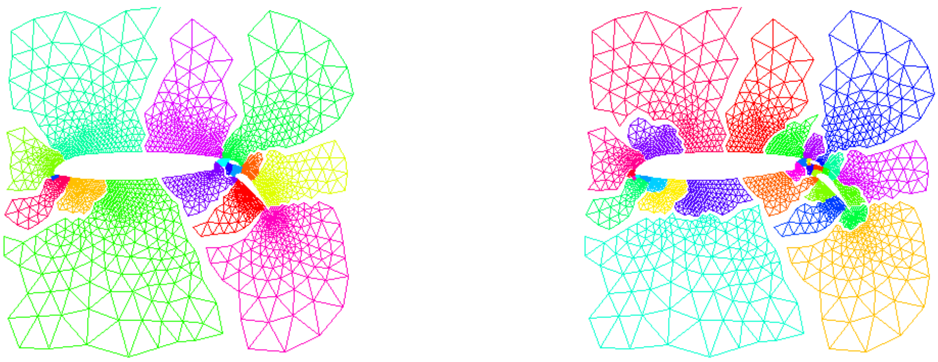


Figure 4. Partitioning results using k-way bisection based on Metis 5.0.2 with $k = 16$ (left) and $k = 32$ (right)

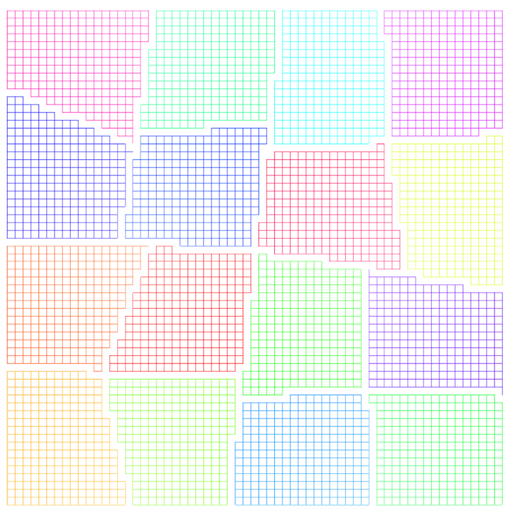


Figure 5. 2D Mesh Partitioning Strategies.

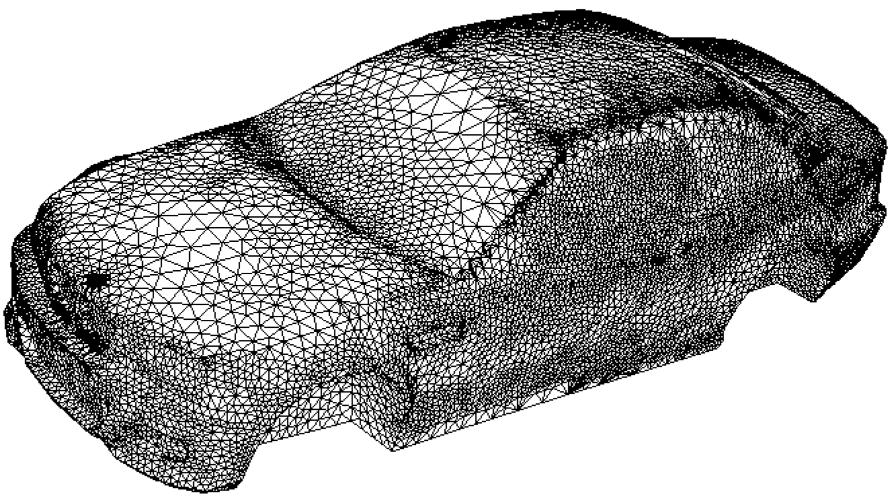


Figure 6. BMW finite element mesh