Enrico Nobile
Dipartimento di Ingegneria e Architettura
Università degli Studi di Trieste

Corso di Termofluidodinamica Computazionale

Homework No. 2
AA 2022/2023

Marzo 2020

Proposed problem

In the Finite Volume method (FV) for general unstructured grids, with cells composed of arbitrary convex polyhedrons, it is convenient to make use of a *face-based*, rather than a *cell-based*, data structure [1]. In this way it is possible to compute convective and diffusive fluxes, for each face, just once, and add them, similarly to the *assembly* process for the Finite Element (FE) method, in the corresponding equations for the two cells who shares the face (edge in 2D), if it is an *internal face*, or otherwise in the equation for the adjacent cell, if it is a *boundary face*.
In addition, it is also convenient, for example for the computation of the gradient, to have available the *connectivity matrix* for the cells, *i.e.* to know, for each cell, the list of all adjacent cells.

For the proposed problem, two generic ASCII files are provided for a 2D unstructured grid composed, for simplicity, only by triangles: gridh.nod and gridh.ele. They are available on the *Moodle* page of the class[1]

The file gridh.nod has the following format:
First row: <number of nodes>
Following rows: <node number> <x> <y> <marker>
where <x> and <y> are the coordinates of the nodes (vertices) of the grid.

The file gridh.ele has the following format:
First row: <number of elements>
Following rows: <element number> <i> <j> <k> <marker>
where <i>, <j>, <k> are the element nodes. The <marker>, unused here, may identify *e.g.* different materials et.

Develop, in MATLAB or other language of choice, a program/script that:

1. Reads both files;
2. Determines the edges of the grid;
3. Finds the adjacent (e.g. neighboring) cells for each cell;
4. Computes the area of all cells and the total area of the domain;
5. Computes the coordinates of the centroids of the cells;
6. Computes the coordinates of the centroids of the edges;
7. Produces a graphical overall representation of the grid, together with the number of the cells. *Optional:* for a limited (zoomed) part of the grid, include also vertex and edge numbers[2].

[1] Due to the size of the files, it is inconvenient, and probably useless, to insert them in an Appendix, as already done in the past for smaller grids.

[2] Doing the same for the entire grid would result in a messy and unreadable drawing.

8. Produces a file, named <gridh.edg>, with the following format:
   First row: <number of edges>
   Following rows: <edge number> <n1> <n2> <e1> <e2>
   where, as illustrated in figure 1, <n1> and <n2> are the nodes that identify the edge, while <e1> and <e2> are the elements (cells) attached to that edge, with the rule that <e2> is set to <-1> if it is a boundary edge.

9. Produces a file, named <gridh.nei>, with the following format:
   First row: <number of elements>
   Following rows: <element number> <e1> <e2> <e3> <d1> <d2> <d3>
   where, as shown in figure 2, <e1>, <e2> and <e3> are the adjacent cells, and <d1>, <d2> and <d3> are the (unique) edges of the cell, with the rule that <ek> is set to <-1> if <dk> is on the boundary.

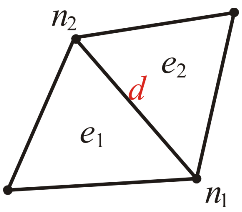

Figure 1: Indication of an edge.

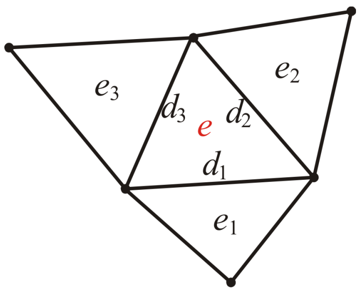

Figure 2: Numbering of edges and neighbor cells.

Note that, in MATLAB, it might be convenient to make use of the function unique and sort to find the *unique* edges of the grid.

References

[1] F. Moukalled, L. Mangani, M. Darwish, *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction With OpenFOAM and Matlab*, Springer Nature, (2015).