Master de sciences et technologie 1 MU4MA029
Mention : mathématiques et applications 2022-2023

TP 2 : mesh management

There are many ways to store the data specifying a mesh. In this tutorial sheet and the following, we shall adopt the following format :

```
$Noeuds
nbr_vtx
num_vtx  x_coord  y_coord
...

$FinNoeuds
$Elements
nbr_elt
num_elt  num_vtx1  num_vtx2  num_vtx3
...

$FinElements
```

This format will allow to store meshes made of triangles. By convention the indices attached to `num_vtx` and `num_elt` start at 0. In appendix is given an example of such a mesh file, as well as its graphical representation. You shall use the packages `NumPy` and `matplotlib` to answer the exercises that follow.

Exercice 1

Question 1.1 Define a function `PrintFile` that takes as input the name of a file (a string variable `str`) and plots in the terminal all the lines of this file.

Question 1.2 Define a function `LoadVTX` that takes as input argument the name of a mesh file (a variable `str`) and returns as output an array of `float` of size `(nbr_vtx)`$\times$ `2` that models the vertices of the mesh. Here and in the following, the mesh file is assumed to comply with the format described above, so that a vertex of the mesh is modelled by an array of two `float` (abscissa and ordinate). One shall for example use the function `split` to "parse" the lines of the mesh file.

Question 1.3 Define a function `LoadELT` that takes the name of a mesh file (a string variable `str`) as input argument, and returns as output an array of `int` of size `(nbr_elt)`$\times$ `3` modelling the triangles of the mesh. Warning : the indices in the arrays start at zero.

Exercice 2

Define a function `GenerateMesh` that generates a mesh file complying with the format described above and modelling a rectangular domain. This mesh is made of rectangles cut in two pieces along a diagonal, like for the mesh in appendix. The function to be written will take as input arguments :

- a `str` : the name of the output file
- a `int` : number of horizontal subdivisions
- a `int` : number of vertical subdivisions
- a `float` : horizontal length of the domain
- a `float` : vertical length of the domain

Exercice 3

Question 3.1 Define a function `PlotMesh` that produces a graphical display of the mesh (see `triplot` in matplotlib) similar to the picture shown in appendix. This function shall take as input :

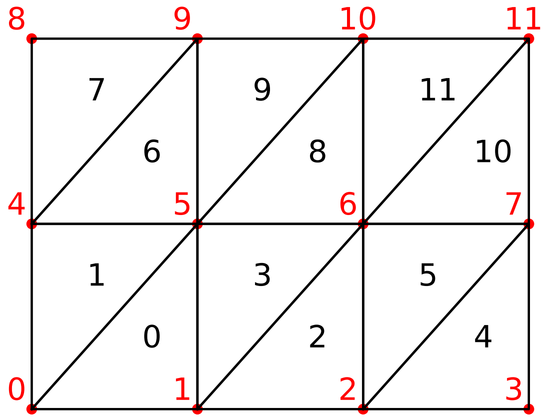
- `vtx` : an array of `float` of size `nbr_vtx`$\times$2 modelling the vertices of the mesh.
- `elt` : an array of `int` of size `nbr_elt`$\times$3 modelling the elements of the mesh.

Question 3.2 Refactor the function `PlotMesh` so that it accepts an overload with, as additional input argument, an array `val` of `float` of size `nbr_vtx`. This array `val` models the nodal values of a function that `PlotMesh` shall graphically plot by means of linear interpolation with a colormap.

Question 3.3 Given a vector $\boldsymbol{d} \in \mathbb{R}^2$ satisfying $|\boldsymbol{d}| = 1$, plot the function $f : \mathbb{R}^2 \to \mathbb{R}$ defined by $f(\boldsymbol{x}) := \cos(4\pi\, \boldsymbol{d} \cdot \boldsymbol{x})$ on the mesh `maillage2.msh` by means of `PlotMesh`.

Appendix : file maillage1.msh

```
$Noeuds
12
0   0.0  0.0
1   1.0  0.0
2   2.0  0.0
3   3.0  0.0
4   0.0  1.0
5   1.0  1.0
6   2.0  1.0
7   3.0  1.0
8   0.0  2.0
9   1.0  2.0
10  2.0  2.0
11  3.0  2.0
$FinNoeuds
$Elements
12
0   0   1   5
1   0   5   4
2   1   2   6
3   1   6   5
4   2   3   7
5   2   7   6
6   4   5   9
7   4   9   8
8   5   6   10
9   5   10  9
10  6   7   11
11  6   11  10
$FinElements
```




```
vtx =
[[0.0,  0.0]
  [1.0,  0.0]
  [2.0,  0.0]
  [3.0,  0.0]
  [0.0,  1.0]
  [1.0,  1.0]
  [2.0,  1.0]
  [3.0,  1.0]
  [0.0,  2.0]
  [1.0,  2.0]
  [2.0,  2.0]
  [3.0,  2.0]]
```

```
elt =
[[0,  1,   5 ]
  [0,  5,   4 ]
  [1,  2,   6 ]
  [1,  6,   5 ]
  [2,  3,   7 ]
  [2,  7,   6 ]
  [4,  5,   9 ]
  [4,  9,   8 ]
  [5,  6,   10]
  [5,  10,  9 ]
  [6,  7,   11]
  [6,  11,  10]]
```