TP 2 : Gestion des maillages

Il existe de nombreuses manières de stocker les données relatives à un maillage. Dans cette séance et celles qui suivront, nous prendrons comme convention le format suivant :

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

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

$FinElements
```

Ce format nous permettra de stocker des maillages formés d'éléments triangulaires. Par convention les numérotations associées à `num_vtx` et `num_elt` commencent à 0. En annexe est donné un exemple d'un tel fichier de maillage, accompagné de sa représentation graphique. Vous pourrez vous aider des packages `NumPy` et `matplotlib` pour répondre aux exercices qui suivent.

Exercice 1

Question 1.1 Definissez une fonction `PrintFile` qui prend en argument le nom d'un fichier (une variable `str` chaîne de caractère) et qui affiche dans le terminal toutes les lignes de ce fichier.

Question 1.2 Definissez une fonction `LoadVTX` qui prend en argument le nom d'un fichier de maillage (une variable `str`) et qui renvoie en sortie un tableau de `float` de taille `(nbr_vtx)`$\times$ `2` représentant les nœuds du maillage. Ici et dans toute la suite, le fichier de maillage est supposé suivre le format décrit plus haut, de sorte qu'un nœud est représenté par un tableau de deux `float` (abscisse et ordonnée). On pourra utiliser la fonction `split` par exemple pour "parser" les lignes du fichier de maillage.

Question 1.3 Definissez une fonction `LoadELT` qui prend en argument le nom d'un fichier de maillage (une variable `str`) et qui renvoie en sortie un tableau de `int` de taille `(nbr_elt)`$\times$ `3` représentant les triangles du maillage. Attention : la numérotation des tableaux commence à zéro.

Exercice 2

Définissez une fonction `GenerateMesh` qui crée un fichier de maillage respectant le format décrit plus haut et associé à un domaine rectangulaire. Le maillage créé par cette fonction est constitué de rectangles coupés en deux selon leur diagonale, comme pour le maillage représenté en annexe. La fonction à écrire prendra en argument :

- un `str` : nom du fichier de sortie
- un `int` : nombre de subdivisions selon l'horizontal
- un `int` : nombre de subdivisions selon la verticale
- un `float` : longueur horizontale de domaine
- un `float` : longueur verticale de domaine

Exercice 3

Question 3.1 Définissez une fonction `PlotMesh` qui réalise un affichage graphique représentant le maillage (voir `triplot` dans matplotlib) comparable à l'image figurant en annexe. Cette fonction prendra en argument d'entrée :

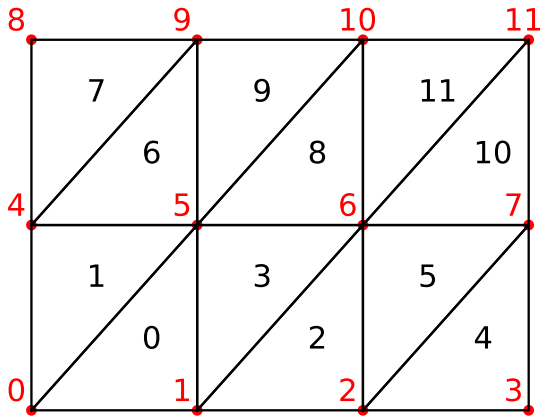
- `vtx` : un tableau de `float` de taille `nbr_vtx`$\times$`2` représentant les sommets du maillage.
- `elt` : un tableau de `int` de taille `nbr_elt`$\times$`3` représentant les éléments du maillage.

Question 3.2 Reprenez la fonction `PlotMesh` pour autoriser une surcharge prenant comme argument supplémentaire une liste `val` de `float` de taille `nbr_vtx`. La liste `val` représente les valeurs nodales d'une fonction dont `PlotMesh` représentera graphiquement l'interpolation linéaire au moyen d'une carte de couleur.

Question 3.3 Étant donné un vecteur $\boldsymbol{d} \in \mathbb{R}^2$ vérifiant $|\boldsymbol{d}| = 1$, réalisez un affichage graphique sur le maillage `maillage2.msh` de la fonction $f : \mathbb{R}^2 \to \mathbb{R}$ définie par $f(\boldsymbol{x}) := \cos(4\pi\, \boldsymbol{d} \cdot \boldsymbol{x})$ en utilisant `PlotMesh`.

Annexe : fichier 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]]
```