

- [TP2–Surfaces de Subdivision](#)
  - [Exercice 1. Maillages triangulaires –Structure de données; Lecture/affichage](#)
  - [Exercice 2. Subdivision de Loop](#)
  - [Exercice 3 Subdivision Butterfly Modifiée](#)

## TP2–Surfaces de Subdivision

---

Vous retrouverez l'énoncé dans ce [fichier](#).

### Exercice 1. Maillages triangulaires – Structure de données; Lecture/affichage

---

Vous retrouvez le code dans le fichier [Exercice1.py](#).

- 1. *Quelle est la structure de données que vous allez utiliser pour représenter les maillages triangulaires 3D?*

Pour représenter les maillages triangulaires 3D, nous allons utiliser une liste de sommets et une liste de triangles. Chaque sommet est représenté par un tuple de trois éléments (x, y, z) et chaque triangle est représenté par un tuple de trois indices de sommets.

- 2. *Écrire une fonction python `Load_off(file_OFF)` et/ou `Load_Obj(file_Obj)` qui lit respectivement un fichier .off et un fichier .obj et l'affiche à l'écran*
- 3. *Écrire une fonction Python `Visualize3DMesh(Vertices, Triangles)` qui visualise un maillage 3D représenté par un ensemble de sommets Vertices et sa topologie (Triangles)*

### Exercice 2. Subdivision de Loop

---

Vous trouverez le code dans le fichier [Exercice2.py](#).

- 1. *Rappeler le principe de la subdivision de Loop*

La subdivision de Loop est une méthode de subdivision de maillage qui consiste à subdiviser chaque triangle en quatre triangles plus petits. Pour chaque sommet, on

calcule un nouveau sommet qui est la moyenne des sommets voisins. On ajoute ensuite un sommet au milieu de chaque arête.

- 2. On considère un maillage manifold fermé (sans bord). a. Écrire une fonction Python qui implémente le schéma de subdivision de Loop pour ce type de maillage b. Afficher le résultat de subdivision pour différentes itérations.
- 3. On considère maintenant un maillage manifold avec bord. a. Que faut-il modifier pour appliquer le schéma de subdivision de Loop? b. Mettre à jour la fonction Python de la question 2. pour tenir compte des maillages avec bord. c. Afficher le résultat de subdivision sur un maillage avec bord pour différentes itérations

## Exercice 3 Subdivision Butterfly Modifiée

---

Vous trouverez le code dans le fichier [Exercice3.py](#).

- 1. Rappeler le principe de la subdivision Butterfly modifiée

La subdivision Butterfly modifiée est une méthode de subdivision de maillage qui consiste à subdiviser chaque triangle en trois triangles plus petits. Pour chaque sommet, on calcule un nouveau sommet qui est la moyenne des sommets voisins. On ajoute ensuite un sommet au milieu de chaque arête.

- 2. On considère un maillage manifold fermé (sans bord).
  - a. Rappeler la notion de valence d'un sommet. Qu'est-ce qu'un maillage régulier?

La valence d'un sommet est le nombre de sommets voisins à un sommet donné. Un maillage régulier est un maillage dont tous les sommets ont la même valence.

  - b. Écrire une fonction Python qui implémente le schéma de manifold pour ce type de maillage.
  - c. Afficher le résultat de subdivision pour différentes itérations.

- 3. On considère maintenant un maillage manifold avec bord.

- a. Que faut-il modifier pour appliquer le schéma de Butterfly modifiée?

Pour appliquer le schéma de Butterfly modifiée à un maillage avec bord, il faut traiter différemment les points sur le bord. Pour ces points, on utilise une formule différente pour calculer leur nouvelle position.

```
for vertex in range(len(vertices)):
    n = len(neighbors[vertex])
    if n > 3: # Point intérieur
        beta = 1/n * (5/8 - (3/8 + 1/4 * np.cos(2*np.pi/n))**2)
        new_vertices[vertex] = (1 - n*beta)*vertices[vertex] +
        beta*sum(new_vertices[neighbor] for neighbor in neighbors[vertex])
    else: # Point sur le bord
        new_vertices[vertex] = 3/4 * vertices[vertex] + 1/8 *
        sum(new_vertices[neighbor] for neighbor in neighbors[vertex])
```

b. \*Mettre à jour la fonction Python de la question 2. pour tenir compte des maillages avec bord.\*

Pour mettre à jour la fonction Python de la question 2 pour tenir compte des maillages avec bord, nous devons ajouter une condition pour vérifier si un sommet est sur le bord du maillage. Si c'est le cas, nous utilisons une formule différente pour calculer sa nouvelle position

Vous retrouver la fonction dans le fichier [Exercice3.py](#).

c. \*Afficher le résultat de subdivision sur un maillage avec bord pour différentes itérations.\*