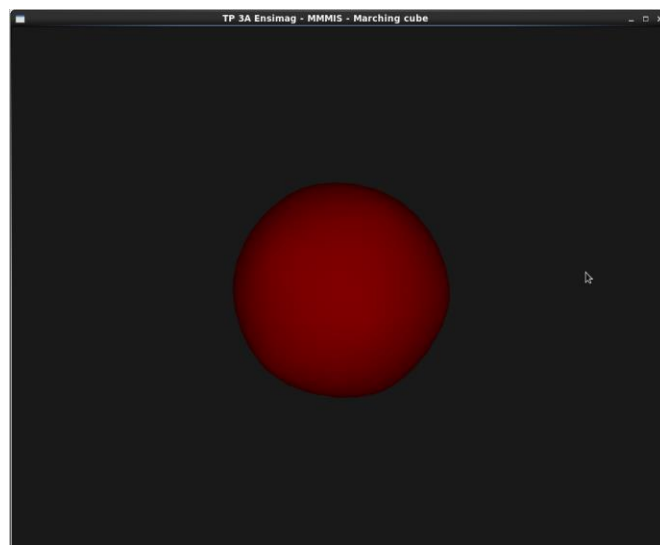
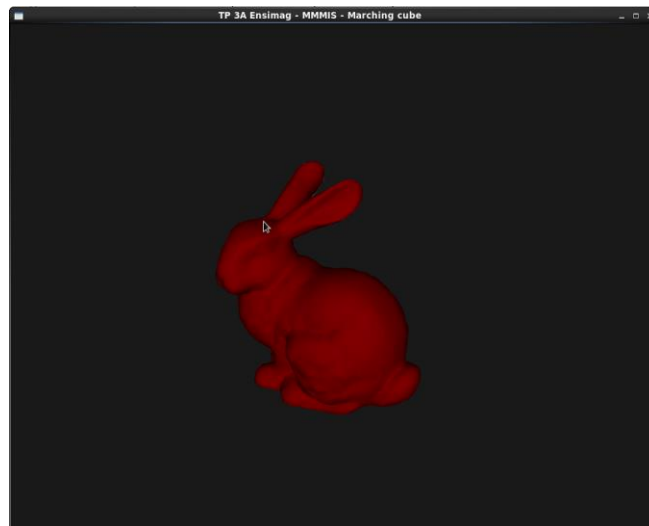
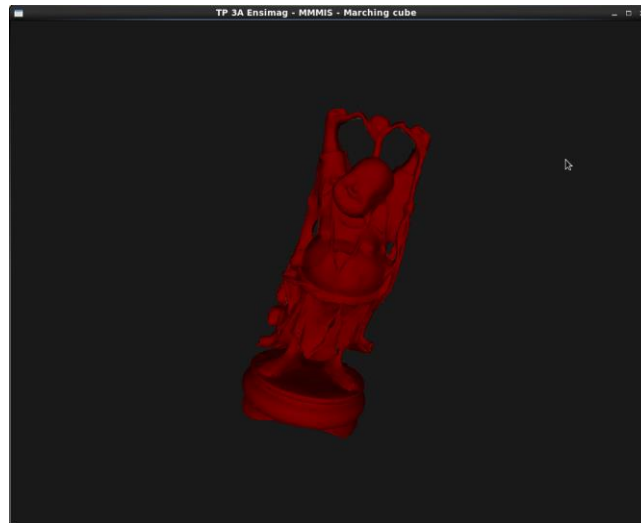


Lissage Laplacien

Henry Lefèvre – M2 Mia Parcours GICAO
Aymeric Seguret – M2 Mia Parcours GICAO



Nous avons testés nos algorithmes sur les différentes mesh suivant :



Démarche du lissage du mesh :

1 - calcul du voisinage

Le voisinage d'un point est calculé à l'aide des attributs twin et next des half_edge. A partir du half edge (d'un vertex donné), le vertex origine de son twin appartient au voisinage. Après ajout de ce point dans la liste des voisins on prend le next de ce twin et on itère jusqu'à retomber sur le half_edge de départ.

2 - calcul du Laplacien

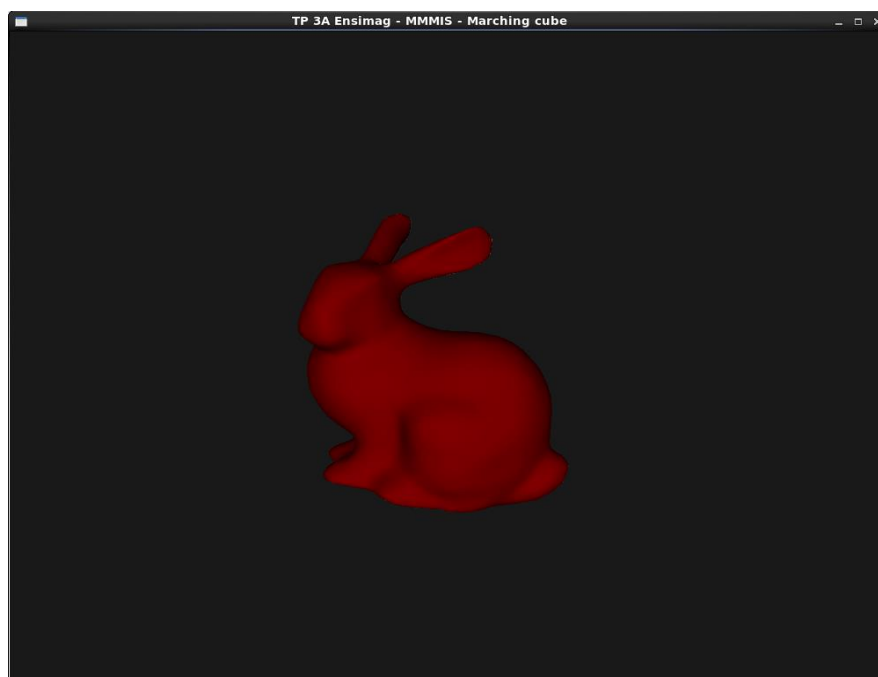
Le Laplacien d'un point du mesh est une direction qui se calcule comme la somme des directions de chaque voisin pondérée par l'inverse du nombre de voisin du vertex considéré. Autrement dit cela représente un déplacement moyenné vers le voisinage de ce point.

3 - lissage itératif

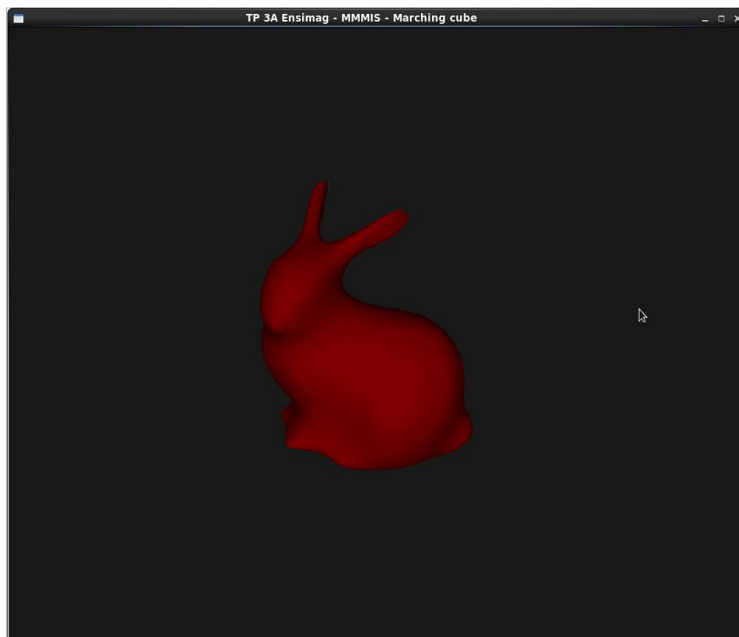
Comme dit dans le TP, le lissage consiste à déplacer chaque point dans la direction de son laplacien pondéré par un coefficient. Ce processus est répété un nombre fixé de fois.

Résultats :

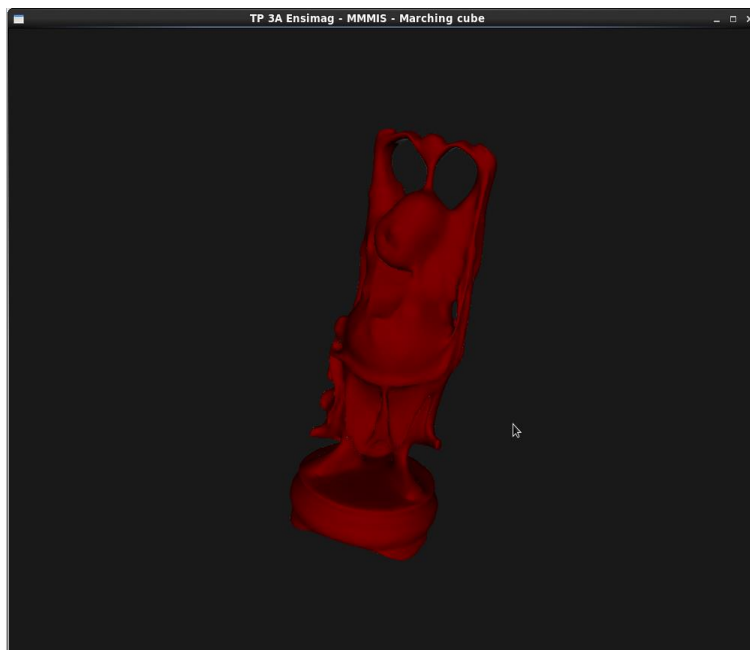
$\lambda = 0.2$, Iter = 20



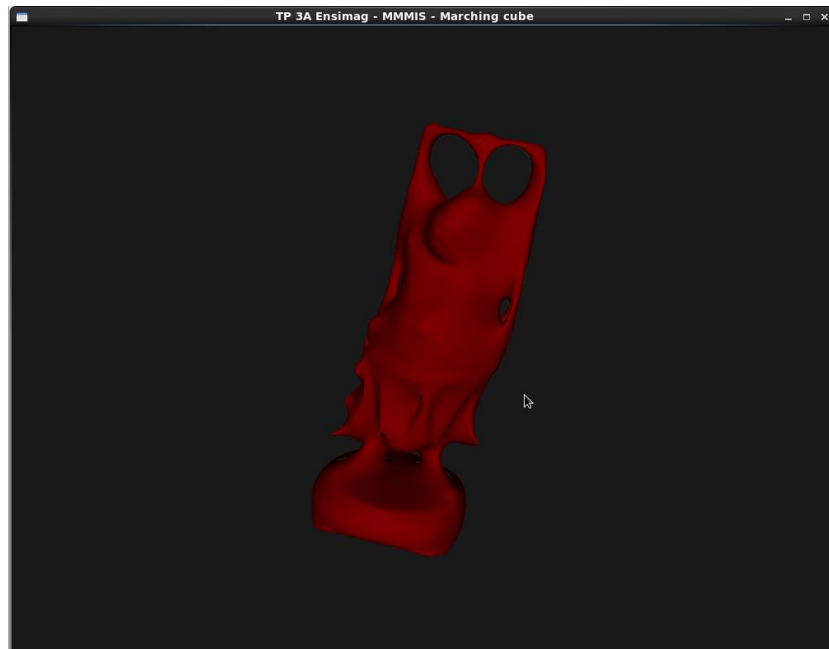
$\Lambda = 1$, Iter = 20



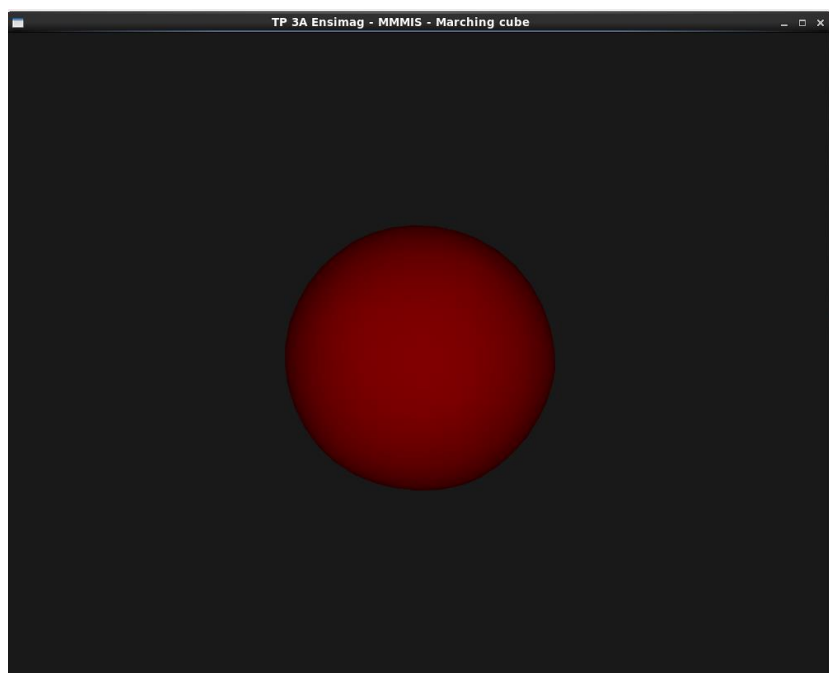
$\Lambda = 0.2$, Iter = 20



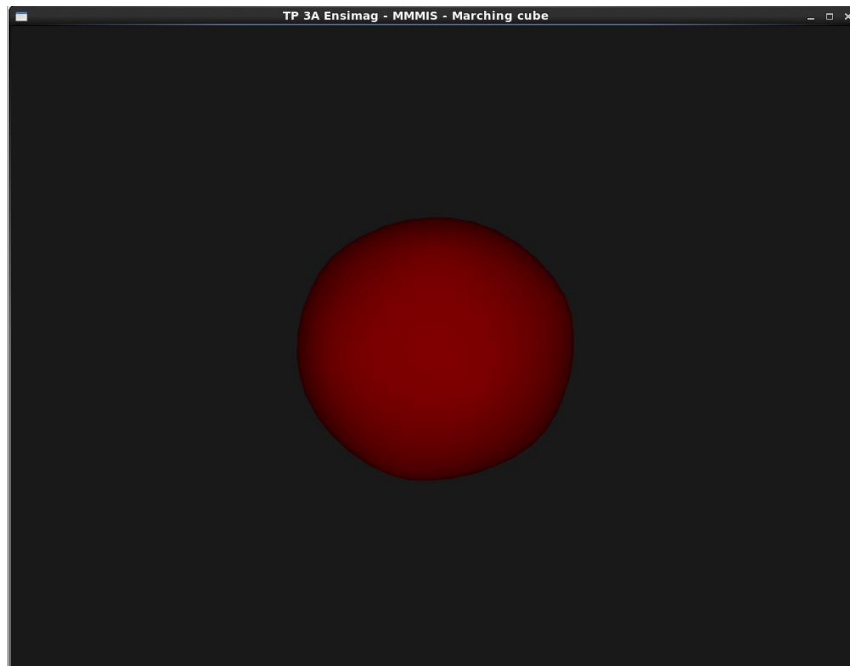
$\Lambda = 1$, Iter = 20



$\Lambda = 0.2$, Iter = 20



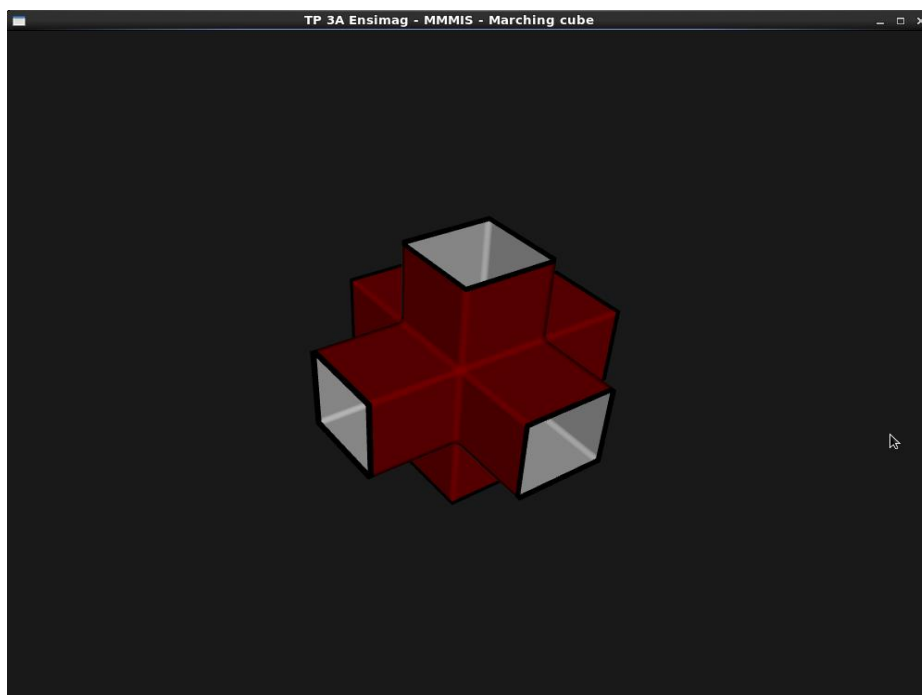
$\Lambda = 1$, Iter = 20



Bonus :

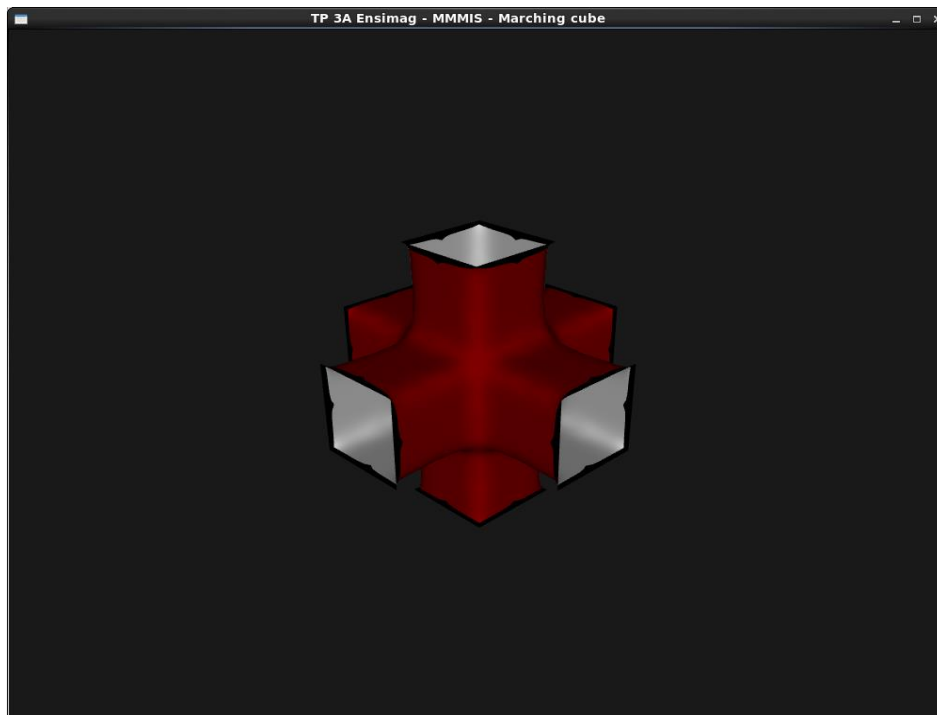
Pour calculer le voisinage d'un vertex qui est au bord, nous voulions repartir du principe de la fonction de voisinage en gérant le cas où l'attribut `twin` d'un `half_edge` est null. Par manque de temps, on a considéré que le voisinage d'un point qui est au bord est égal au point lui-même.

Ainsi avec des surfaces non fermées, par exemple celle dessous :



Nous obtenons pour un lissage Laplacien :

$\lambda = 0.8$, Iter = 10



Et pour un lissage de Taubin :

$\lambda = 0.330$, $\mu = -0.331$, Iter = 30

