

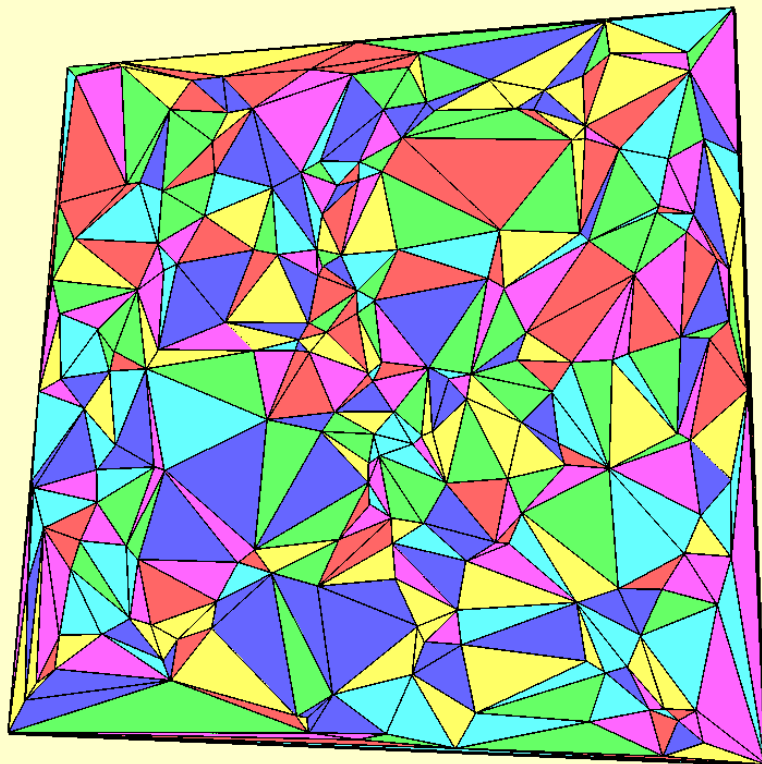


ÉCOLE
CENTRALE LYON

Colin CROS

RAPPORT DE BE

Mesh and Computational Geometry



Année scolaire 2019-2020
Professeur : Raphaëlle Chaine

Introduction

Ce rapport présente les différents rendus obtenus lors des BE du cours de Mesh and Computational Geometry. Les captures d'écran ont été réalisées sur les sorties du programme QT Creator implémenté pendant les BE (figure 1).

L'ensemble des codes du programme est disponible dans un répertoire github accessible à l'adresse : <https://github.com/TastyColin/Mesh-and-Computational-Geometry/>. Les fichiers des maillages utilisés (au format "off") sont également disponibles à cette adresse.

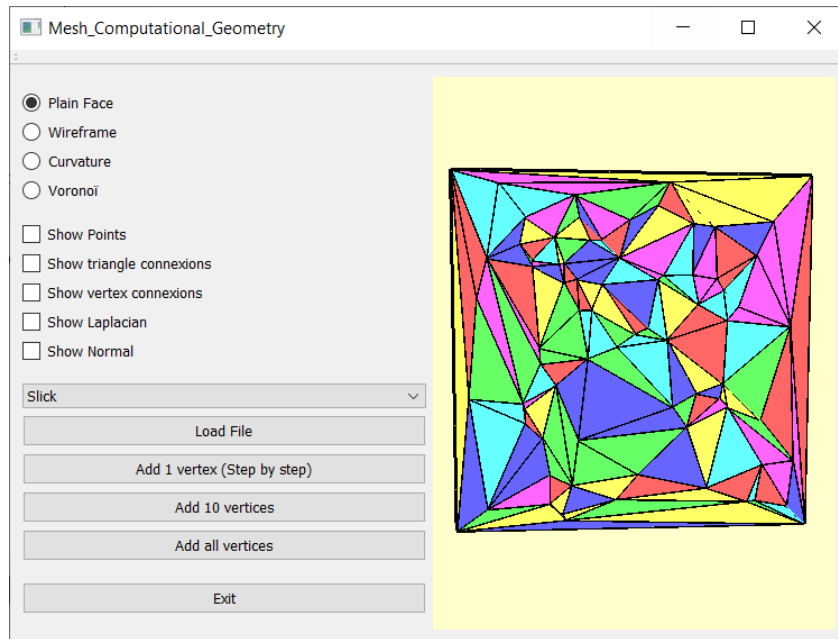


FIGURE 1 – Programme réalisé pendant les BE.

4 modes de visualisation sont disponibles dans le menu du programme qui affichent respectivement (de haut en bas) : les faces, les arêtes, la courbure et le diagramme de Voronoï associé à un fichier. Les "checkboxes" permettent d'ajouter des compléments d'information : les sommets, les différentes liaisons (triangle-triangle ou sommet-triangle), les normales, et le laplacien. Le menu déroulant sert à sélectionner un fichier que l'on importe grâce au bouton "Load File". Les 3 boutons d'ajout permettent de compléter la triangulation (uniquement sur le fichier "slick.off"). Une animation "pas à pas" a été implémentée lors de l'ajout d'un seul point pour illustrer les phases de recherche de point et de retournement des arêtes.

1 Structure de données

1.1 Les fichiers OFF

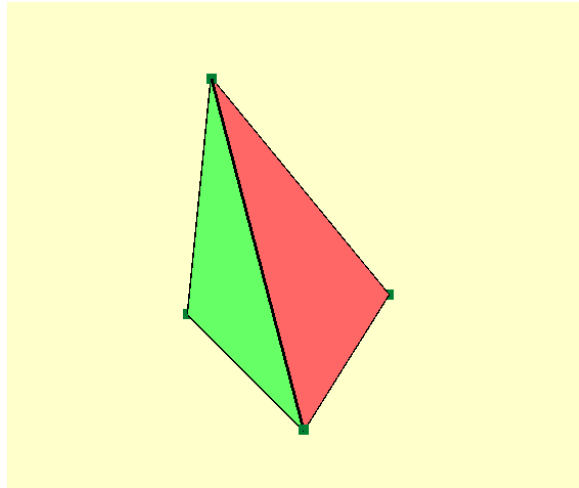
Les fichiers utilisés dans les BE étaient au format "off". Ce format de fichier est composé de trois types de lignes. Tout d'abord, une ligne d'entête indiquant dans cet ordre le nombre de sommets, le nombre de faces et le nombre de trous. Viennent ensuite les lignes indiquant la position x , y , z des sommets (une ligne par sommet). Et pour finir les lignes des faces qui débutent par le nombre de sommets constituant la face suivi de la liste des indices des sommets parcourues dans le sens trigonométrique (lorsque l'on regarde la face de l'extérieur).

Le séparateur utilisé est une espace. A titre d'exemple, voici le code réalisé pour un tétraèdre :

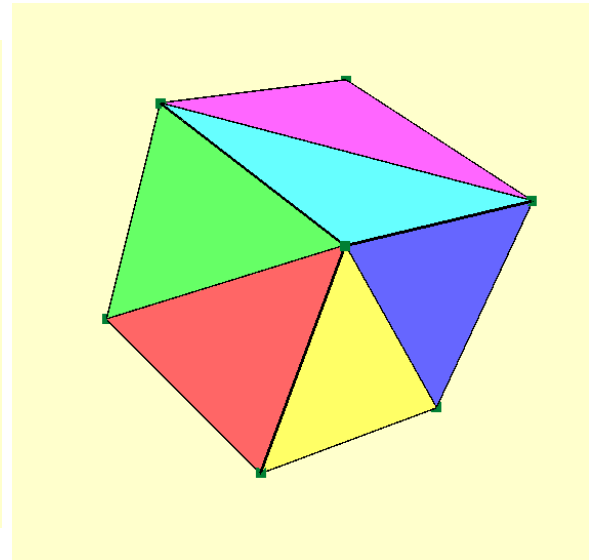
```
1 4 4 0 # nb de sommets, nb de faces, nb de trous
2 0. 0. 0. # sommet 1
3 1. 0. 0. # sommet 2
4 0. 1. 0. # sommet 3
5 0. 0. 1. # sommet 4
6 3 1 2 3 # face 1 (un triangle)
7 3 0 3 2 # face 3 (un triangle)
8 3 0 1 3 # face 2 (un triangle)
9 3 0 2 1 # face 4 (un triangle)
```

Nous ne traiterons que des triangulations, en conséquence, toutes les faces seront triangulaires et sans trou.

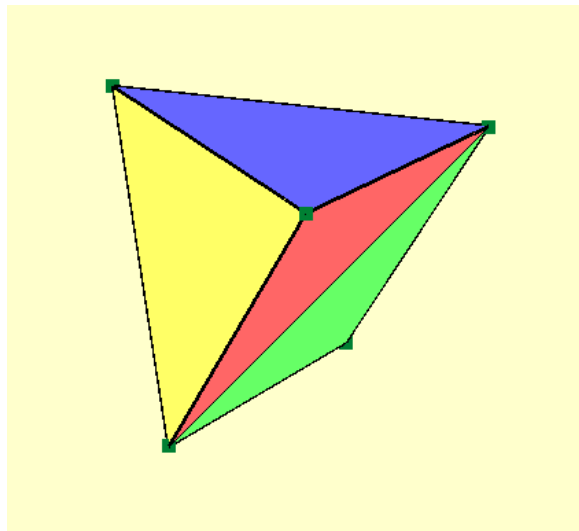
Trois fichiers "off" ont été réalisés : un tétraèdre, un cube et une pyramide à base carrée. Un quatrième fichier a été fourni, il s'agit d'un maillage du visage d'une femme. Les affichages de ces quatre maillages sont visibles sur la figure 2.



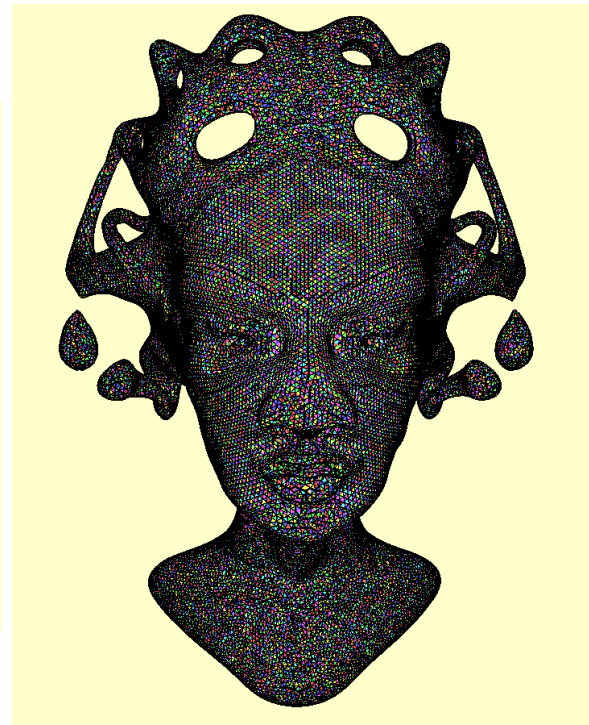
(a) Tétraèdre



(b) Cube



(c) Pyramide à base carrée



(d) Visage

FIGURE 2 – Différents maillages utilisés

L'affichage est réalisé uniquement grâce aux positions des sommets des triangles. A ce stade, il est impossible de tourner autour d'un sommet ou de se déplacer de face en face. Les structures de données doivent être enrichies.

1.2 Enrichissement des structures de données

Deux structures principales composent les maillages : les sommets et les triangles. Les sommets comme les triangles seront stockés dans des vecteurs, on pourra donc les référencer grâce à leur indice.

Les sommets

Les sommets composent les faces des maillages. Un sommet est défini par un point, c'est-à-dire des coordonnées de l'espace, et l'indice d'un triangle qu'il compose. On enrichit la structure avec un second point qui servira à mémoriser le résultat de calculs ultérieurs, par exemple celui du laplacien.

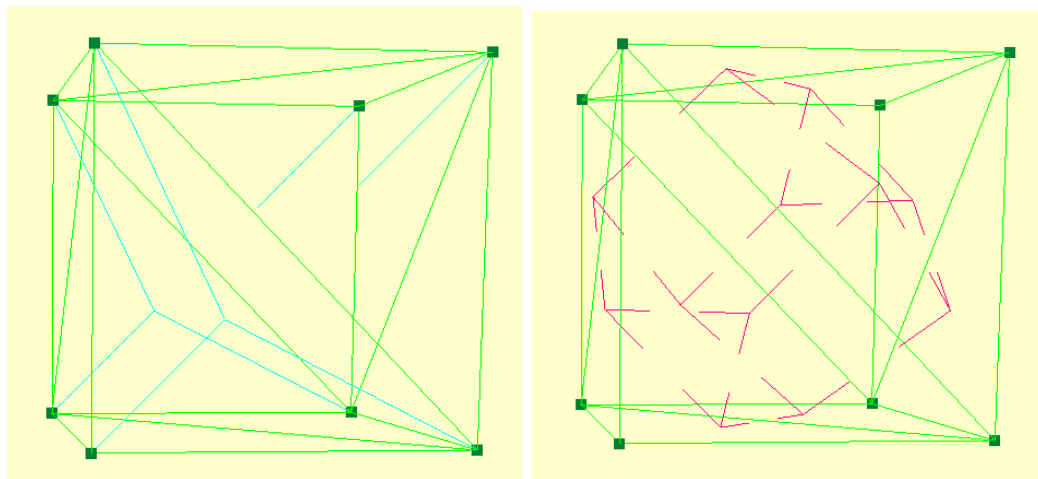
```
1 struct Vertex
2 {
3     Point point;
4     int i_triangle = -1;
5     Point vector_value = Point(0,0,0);
6 };
```

Les triangles

Les triangles sont les éléments élémentaires des triangulations. Sans surprise, un triangle est défini par un triplet de sommets (on enregistrera leur indice), et un triplet d'indices de triangles correspondant à ceux des triangles adjacents.

```
1 struct Triangle
2 {
3     int i_vertices[3];
4     int i_triangles[3];
5 };
```

Pour vérifier la bonne implémentation de ces deux structures de données, un segment représentant l'adjacence entre les triangles est affiché sur la figure. De même un segment reliant un sommet à sa face associée est affichable. La figure 3 affiche ces deux types de liaisons.

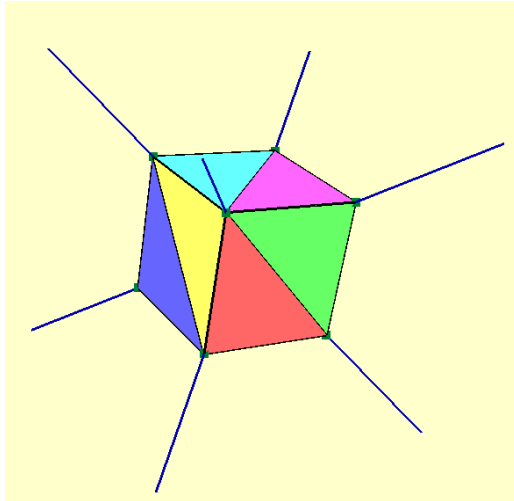


(a) Représentation des connexions entre un sommet et le centre de sa face associée sur un cube. Les segments relient les sommets au centre de leur face associée. (b) Représentation des connexions entre les faces d'un cube. Les segments partent du centre d'un triangle et pointent vers le centre d'une face adjacente.

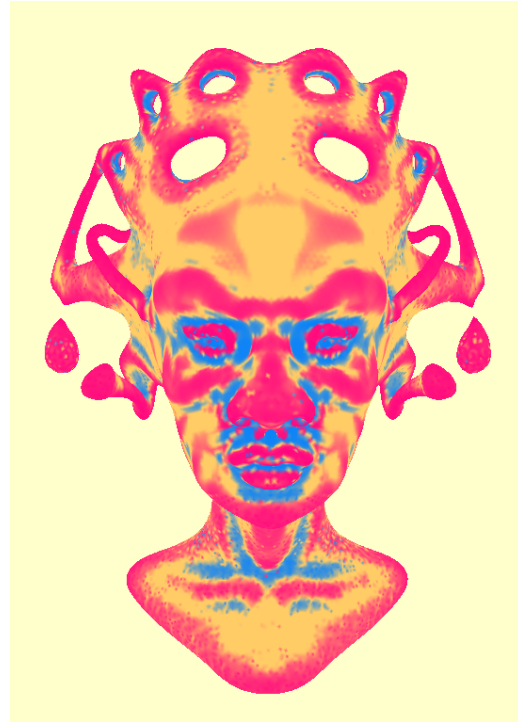
FIGURE 3 – Représentation des différentes liaisons sur un cube

2 Circulateurs et calcul courbure

Dans cette partie, nous allons réaliser le calcul et l’affichage de la courbure. Après avoir implémenté des circulateurs parcourant les faces ou les sommets incidents à un sommet donné. On peut calculer la normale et la courbure à ce sommet. On applique pour ce faire la formule faisant intervenir les cotangentes des angles adjacents qui a été donnée en cours et qui ne sera pas rappelée ici.



(a) Affichage des normales du cube



(b) Affichage des rayons de courbure sur le visage

FIGURE 4 – Affichage de la courbure

Pour représenter les courbures, on affiche l’inverse du laplacien c’est à dire les rayons de courbure. La figure 4b représente les rayons de courbure obtenus sur le maillage de visage. La plage de couleur a été choisie pour afficher en orangé les rayons de courbure grands en valeur absolue (ils correspondent aux zones plates), en rouge les petits rayons de courbures négatifs correspondant à des bosses (zones convexes) et en bleu les petits rayons de courbures positifs correspondant à des creux (des concavités).

3 Triangulation de Delaunay et diagramme de Voronoï

Afin de réaliser des triangulations Delaunay, un 5ème fichier "slick.off" a été ajouté (c.f. figure 5). Il contient une pyramide à base carrée avec 100 points tirés aléatoirement sur sa base. Ce fichier va permettre de réaliser les triangulations de Delaunay.

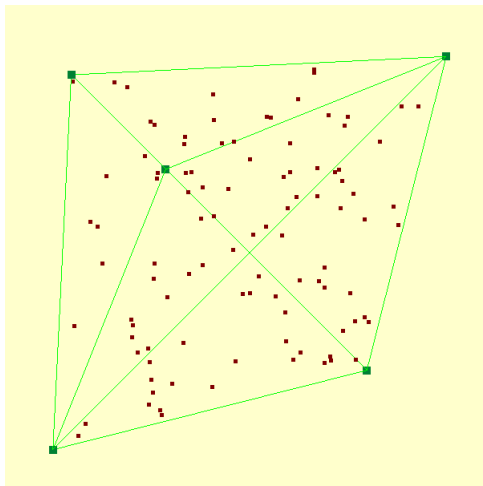
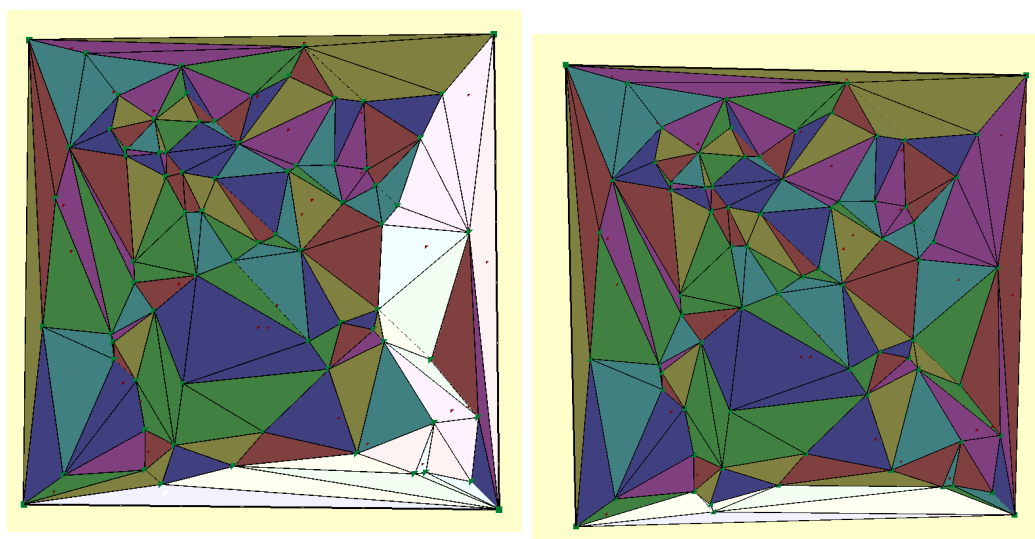


FIGURE 5 – Visualisation du fichier "slick.off". Les points rouges ont été tirés aléatoirement et uniformément sur la base. Initialement, ils n'apparaissent dans aucun triangle.

Pour réaliser la triangulation de Delaunay, les sommets sont ajoutés un à un au maillage en effectuant une marche par visibilité pour trouver le triangle dans lequel ils doivent être insérés. La figure 6a illustre ce processus de recherche. Une fois le sommet inséré, pour maintenir le caractère global de Delaunay, une série de retournements d'arêtes est réalisée autour du nouveau sommet. Les triangles dont les arêtes sont testées lors de la phase de retournement sont illustrés figure 6b.



(a) Mis en lumière de la marche réalisée. Seuls les triangles clairs ont été testés lors de l'étape de flip. L'animation est (en commençant par le triangle le plus à droite).

FIGURE 6 – Insertion d'un sommet

Enfin, à partir de la triangulation de Delaunay, on peut afficher le diagramme de Voronoï as-

socié (figure 7). Pour cela on calcule et enregistre les coordonnées des centres des triangles qui sont nœuds du diagramme.

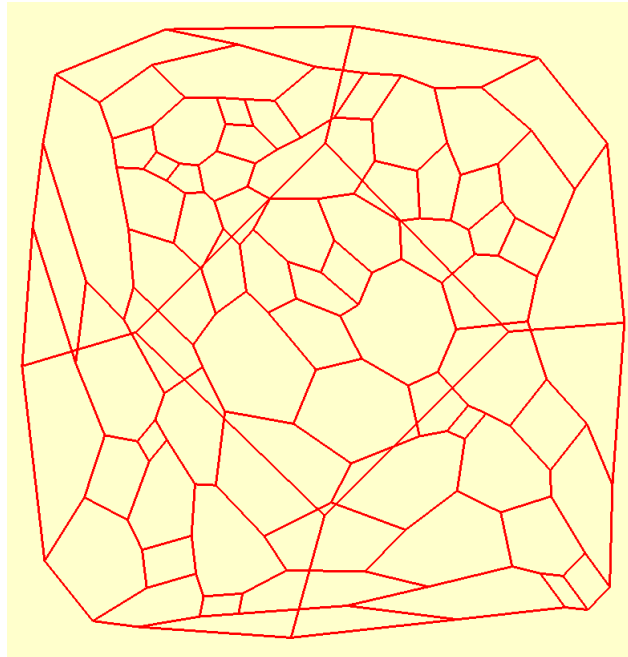


FIGURE 7 – Diagramme de Voronoï associé à la triangulation précédente. Seul les arcs sont représentés, le losange du fond correspond aux faces de la pyramide.

Conclusion

Ces BE ont été l'occasion de comprendre les bases de l'informatique géométrique. En particulier de ce sensibilisé à ses subtilités, je pense notamment aux problèmes d'approximation que l'on néglige souvent mais qui peuvent s'avérer fatals pour certaines applications.

Je souhaiterais m'orienter vers les mathématiques appliquées à la navigation et je pense que les concepts étudiés durant ce MSO me seront utiles, à commencer par le simple fait d'avoir développé le programme en C++.

De manière générale, j'ai apprécié le format du cours : 2h de cours puis 2h de BE (un peu modulé en fonction des séances). Un léger point d'amélioration serait à mon sens le programme fourni au début des BE qui a l'air obsolète. Par ailleurs, il faudrait aussi prévenir avant le premier cours d'installer QT Creator, pour ne pas souffrir de la connexion des salles de centrale.