

Géométrie Algorithmique et Maillage

Compte rendu de TP

Objectif du TP :

Mettre en place un algorithme permettant de lire un maillage depuis un fichier off.
Mettre en place des itérateurs et des circulateurs pour parcourir le maillage.
Mettre en place de calculer le laplacien de l'ensemble des points de la triangulation.
Mettre en place un algorithme permettant d'effectuer la triangulation de delaunay d'un maillage et la génération d'un diagramme de voronoi.
Mettre en place un des algorithmes vu en cours : Simplification de maillage, Crust, Ruppert.

Outils Utilisés :

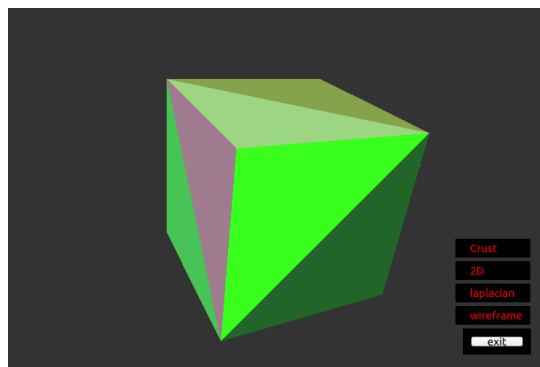
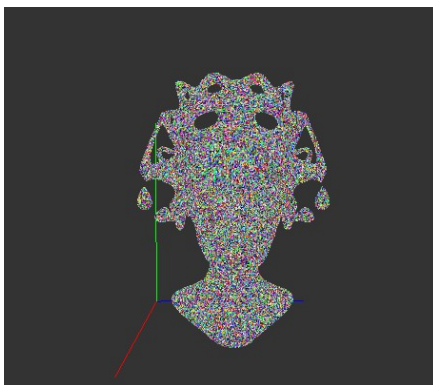
Qt créator

Code réalisé :

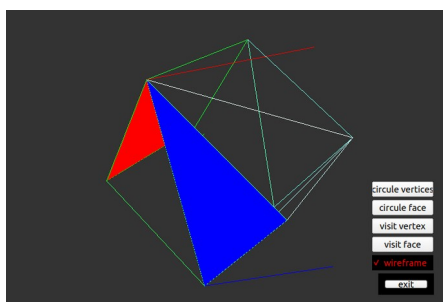
Sur les objectifs du tp nous avons réussi à mettre en place la lecture d'un maillage depuis un fichier off ainsi que l'ensemble des itérateurs et circulateur.

Exemple de résultat :

Lecture fichier off :

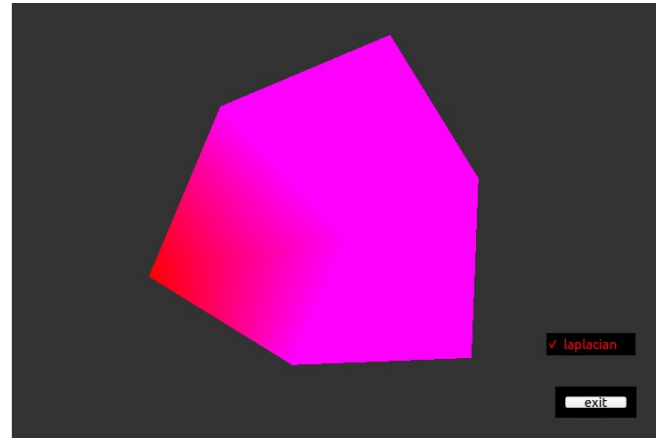
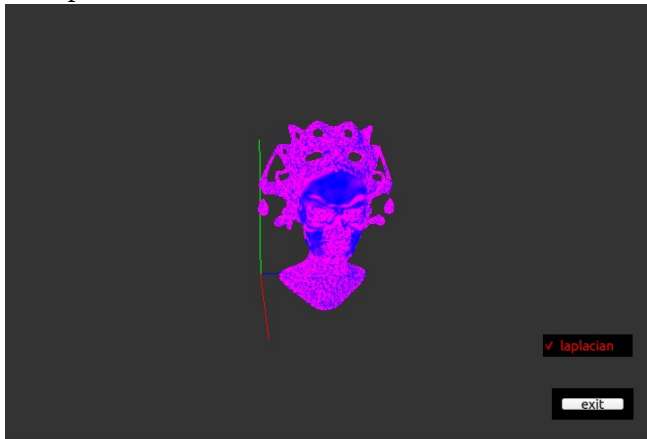


Circulateur (bleu) et itérateur (rouge) :



Malheureusement nous n'avons pas réussi à mettre en place un laplacien satisfaisant. Comme vous pouvez le voir sur l'exemple de la reine bien que les parties plates ai une tendance bleu et les parties courbés tendes vers le rouge (ce qui est attendu) nous avons que très peu de points rouges. On peut en conclure à des valeurs aberrantes qui dévient notre estimation. Comme on peut le voir sur le cube qui devrait être entièrement rouge un point à une valeur plus importante que les autres.

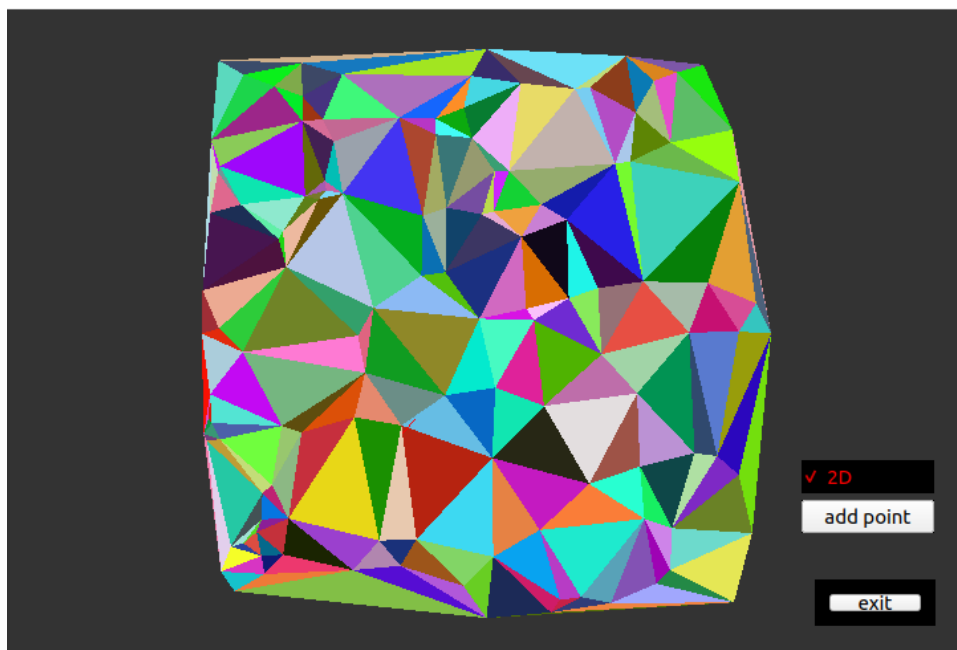
Exemple de résultat :



Nous avons aussi mis en place un algorithme capable de générer un maillage triangulé de delaunay à partir de l'ajout de point et d'un maillage mal triangulé.

Cet algorithme incrémentale parcourt la totalité des triangles à chaque ajout de point pour savoir ou ajouter le point puis analyse les triangles autour de ce point afin de les rendre de delaunay. À chaque fois qu'il transforme un triangle non de delaunay il test les triangles autour du triangle transformé afin de les rendre de delaunay si la modification les a affecté.

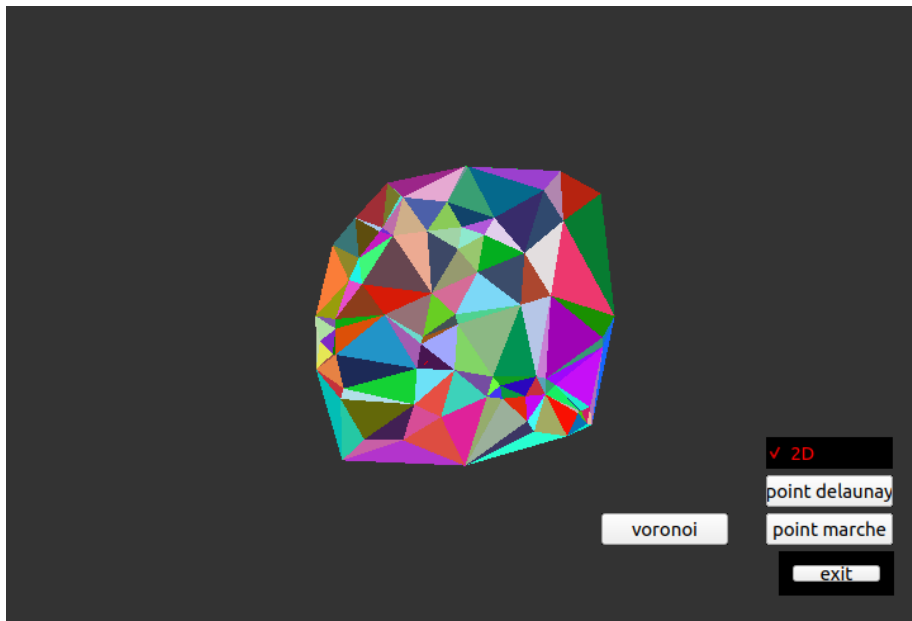
Exemple de résultat :



Nous avons proposé un deuxième algorithme capable de faire la même chose que le premier algorithme.

À la différence du premier algorithme, celui-ci ne parcourt pas tous les triangles à chaque itération. Dans cet algorithme nous prenons l'un des triangles du mesh aléatoirement et nous nous déplaçons dans la direction du point jusqu'à trouver un triangle infini ou un triangle qui contienne le point.

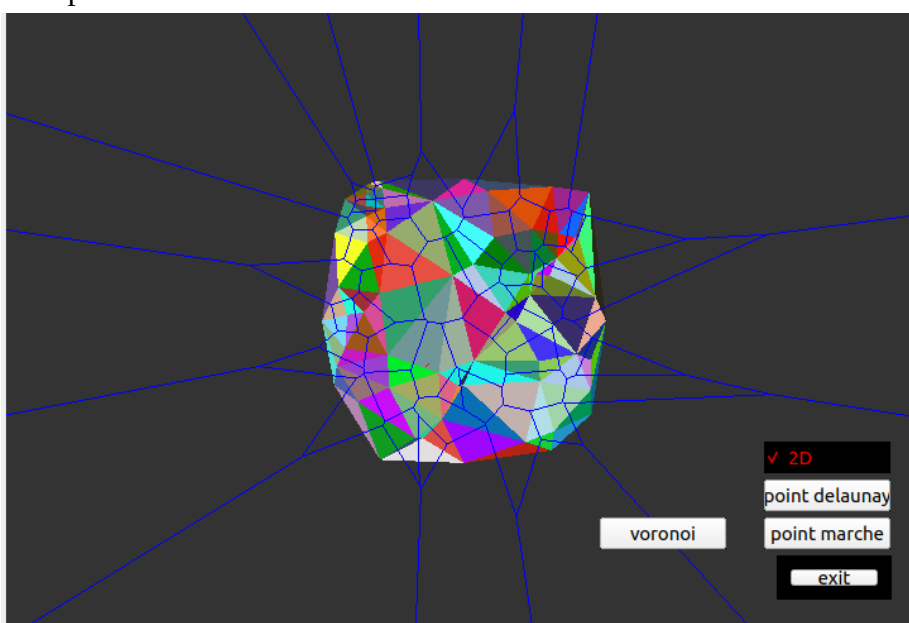
Exemple de résultat :



Après avoir certifié que notre triangulation était correcte, nous avons mis en place un algorithme permettant la création d'un diagramme de voronoi à partir de la triangulation de delaunay obtenue.

Cet algorithme calcule le centre du cercle circonscrit au triangle et définit une cellule de voronoi comme étant l'ensemble des centres circonscrits aux triangles de faces connectées autour d'un point.

Exemple de résultat :



Enfin nous avons choisi de mettre en place l'algorithme de crust nous permettant de détecter le squelette d'un maillage dans une triangulation de delaunay.

Pour cela nous triangulons notre maillage de base puis nous y insérons nos centres de voronoi, une fois cela fait nous obtenons une nouvelle triangulation de delaunay et nous cherchons dedans les segments qui relient deux points de voronoi et qui appartiennent pas a un triangle composé que de points de voronoi.

Exemple de résultat :

