

RAPHAËLLE CHAINE

DÉCEMBRE 2023

Maillage et géométrie algorithmique

RÉDIGÉ PAR

SPATARO Mathis - p1819506

ROULLIER Léa - p1911736

INTRODUCTION

Ce projet a pour but d'implémenter une *bibliothèque simple de chargement et de manipulation de maillages*. Les fonctionnalités que nous avons sont :

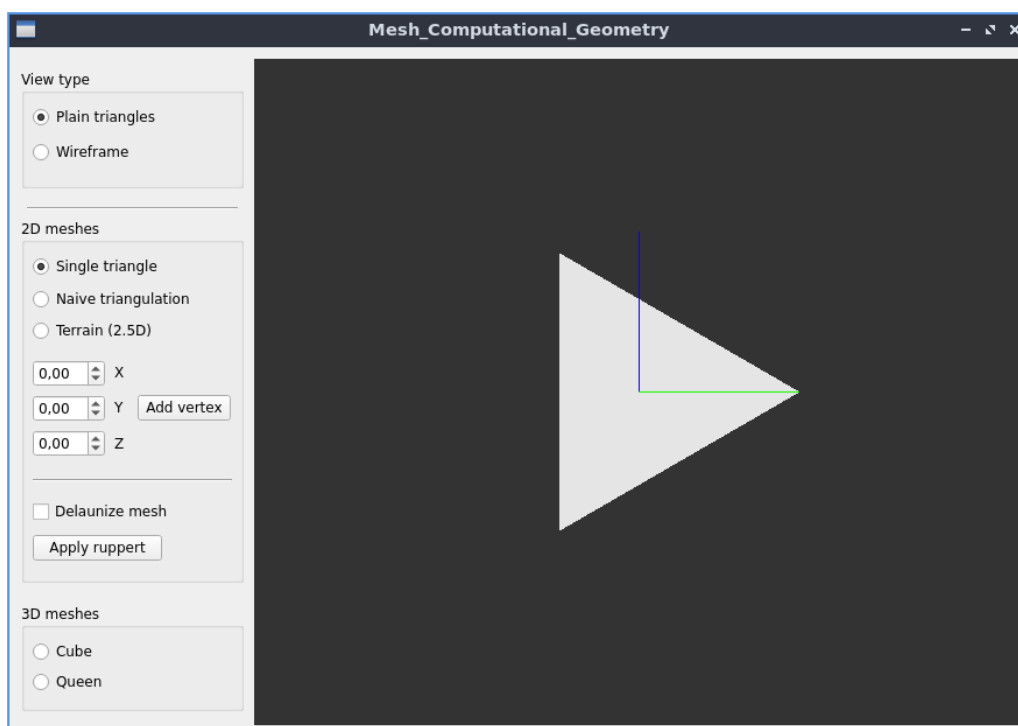
- Le **chargement d'un fichier off**.
- L'**écriture dans un fichier off**.
- Le **chargement d'un jeu de points**.
- L'**ajout de points** dans le maillage 2D en préservant l'enveloppe convexe.
- L'**algorithme de Lawson** d'un maillage 2D.
- L'**algorithme de Ruppert** d'un maillage 2D.

INSTALLATION

Pour *installer* le projet, il faut :

1. **Clôner** le dépôt.
2. Modifier la variable **projectPath** avec le chemin absolu vers la racine du projet. Exemple : `"/home/user/Documents/M2S1/gam/repo/gam-roullier-spataro/"`.

INTERFACE



View type

☒ Plain triangles

☐ Wireframe

View type permet de choisir la **façon dont nous allons voir notre maillage** :

- **Plain triangles** : les triangles sont pleins.
- **Wireframe** : seulement les arêtes sont affichées.

2D meshes

☒ Single triangle

☐ Naive triangulation

☐ Terrain (2.5D)

0,00 X

0,00 Y

0,00 Z

☐ Delaunize mesh

2D meshes permet de **manipuler et visualiser des maillages en 2D** :

- **Single triangle** : affichage d'un triangle (enregistré au format off).
- **Naive triangulation** : affichage d'un maillage crée avec 10 points aléatoires (enregistré au format off).
- **Terrain** : maillage d'un terrain crée à partir du jeu de points "alpes_poisson" fourni en ajoutant les points un par un.
- **Add vertex** : ajoute un sommet aux coordonnées spécifiées et préserve l'enveloppe convexe en 2D (la coordonnée Z n'est pas prise en compte sauf pour l'affichage).

-
- **Delaunize mesh** : rend le mesh présent à l'écran delaunay.
 - **Apply ruppert** : rend un mesh de delaunay ruppertisé (ne fonctionne que si le mesh est de delaunay et en 2D).

3D meshes

☐ Cube

☐ Queen

3D meshes permet de **charger des fichiers au format off** :

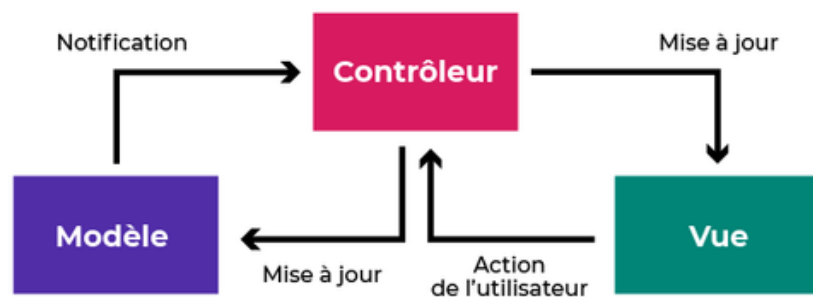
- **Cube** : .off fourni.
- **Queen** : .off fourni.

Nous avons implémentés **plusieurs controles** :

- **Clic + déplacement souris** : tourner la caméra.
- **Molette** : zoom.
- **z, q, s et d** : déplacer la caméra.

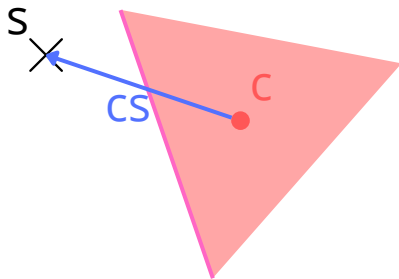
IMPLÉMENTATION

- Un maillage est composée d'une liste de **Face**, de **Vertex** et de **laplaciens**.
- Une Face est une liste de **trois Vertex** et de **trois Face voisines**. L'indice i d'une Face correspond à l'**indice du Vertex opposé**. Les sommets d'une face sont rangés dans l'ordre **trigonométrique**.
- Deux types de structures permettent de parcourir le maillage : **les itérateurs et les circulateurs**. Les listes sont implémentées par des **"vectors" de la STL**, les itérateurs de faces ou de sommets sont donc des itérateurs standards sur des vectors. Nos circulateurs sont **bidirectionnels**, ils circulent sur les sommets ou sur les faces autour d'un sommet donné et s'arrêtent s'ils sont au **bord du maillage**.
- Les **arêtes** sont définies soit par l'**indice d'un triangle et l'indice du sommet opposé à l'arête** dans ce triangle ou par les **indices de deux triangles**, ou encore par les **indices de deux sommets**.
- Le programme est découplé en 3 parties selon un pattern **modèle - vue - contrôleur** et la vue ne se met à jour qu'à travers une interface ne permettant que la lecture, sur signal du contrôleur qui agit comme intermédiaire entre l'interface utilisateur et le reste du programme.

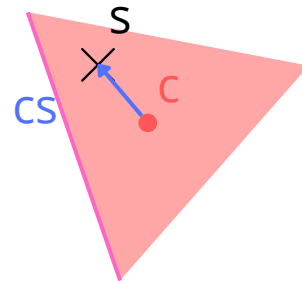


- Le **monde** est implémenté comme un singleton mais pas le mesh, permettant éventuellement d'**étendre le programme à plusieurs meshes**.
- Pour représenter l'inexistence ou la suppression d'un élément, on utilise par convention la valeur **UINT_MAX** correspondant au nombre maximal représentable par un entier non signé.

- Pour la **recherche de la face la plus proche** d'un sommet (**s**), on calcule le centre de la face courante (**c**) ainsi que le vecteur du centre vers le sommet (**cs**). Ensuite on trouve l'**arête du triangle qui intersecte ce vecteur** (**_**), s'il y a un triangle de l'autre côté de l'arête on continue sinon on s'arrête.

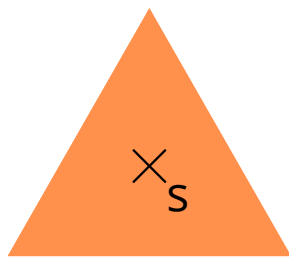


Terminaison 1 : pas de prochain triangle.

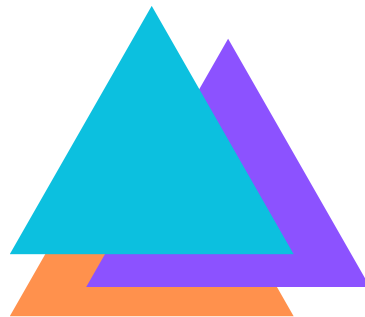


Terminaison 2 : le sommet est dans le triangle.

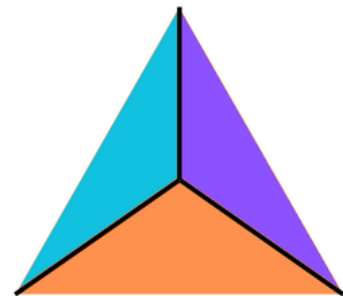
- Pour le **découpage (split) d'une face**, on a un triangle où l'on souhaite ajouter un sommet (étape 1), on **copie ensuite le triangle deux fois** (étape 2) puis on **remplace un sommet par triangle** (y compris le triangle original) **par le nouveau sommet** (étape 3). Ensuite on ajuste les voisins pour respecter la topologie.



Étape 1



Étape 2

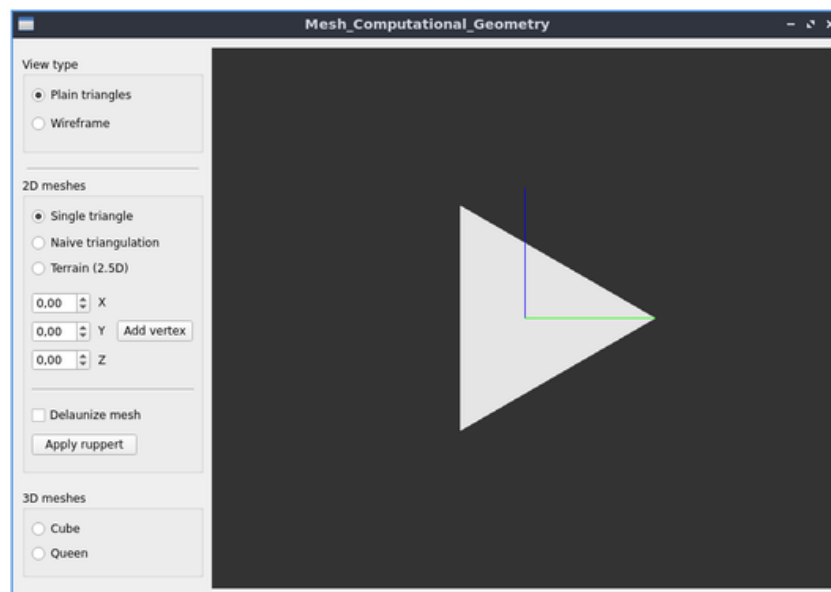


Étape 3

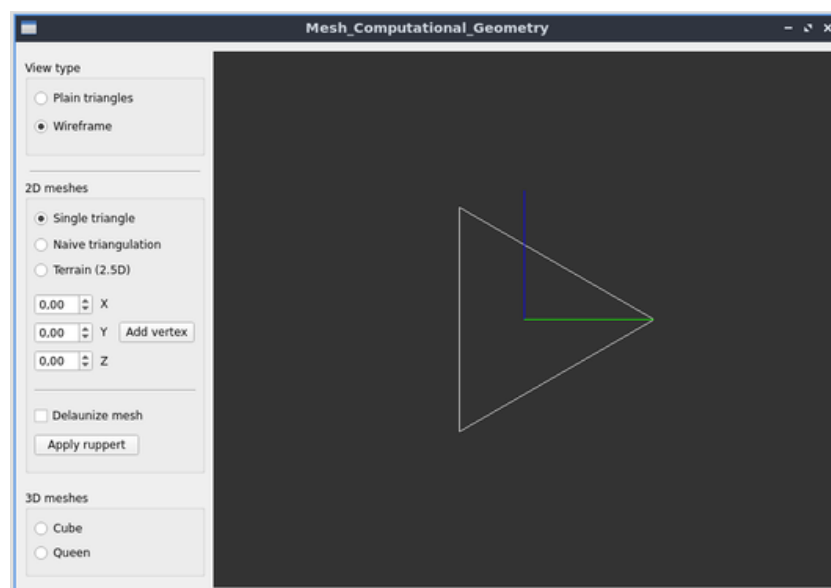
RÉSULTATS

VIEW TYPE

Triangle plein

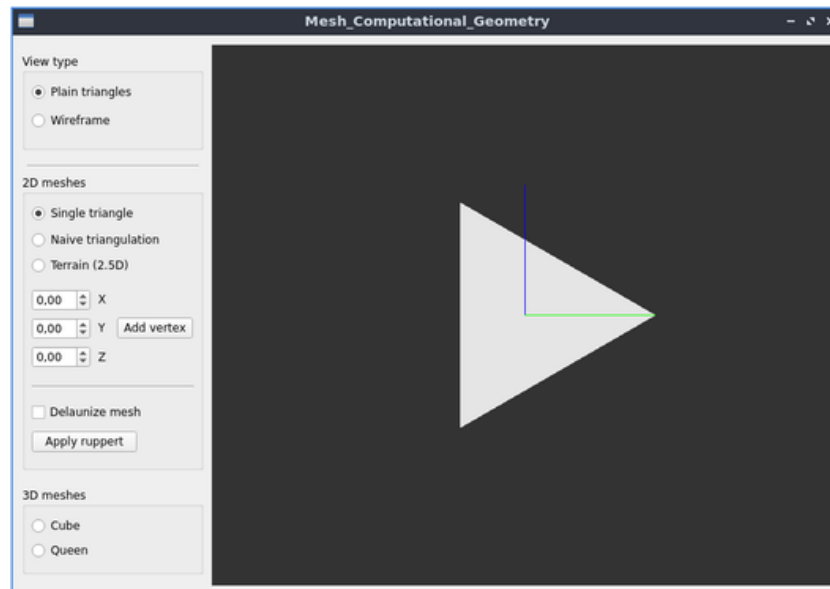


Triangle creux

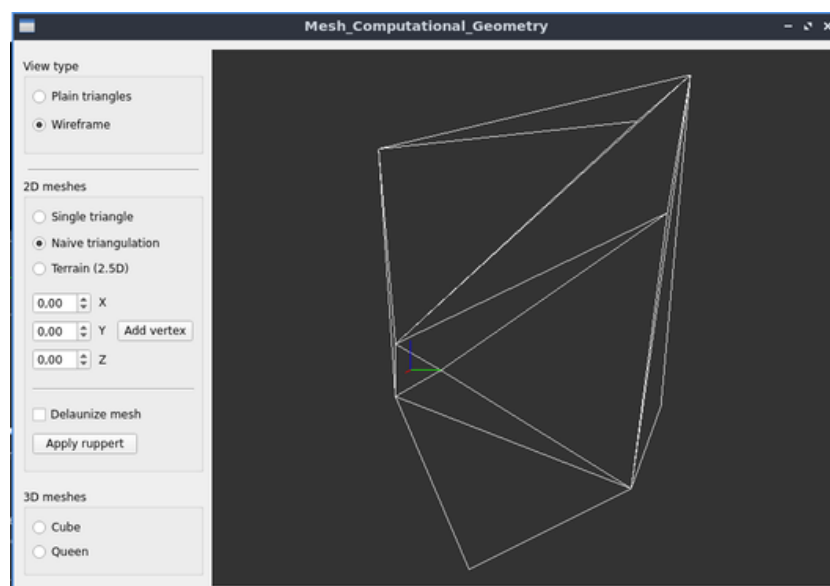


2D MESHES

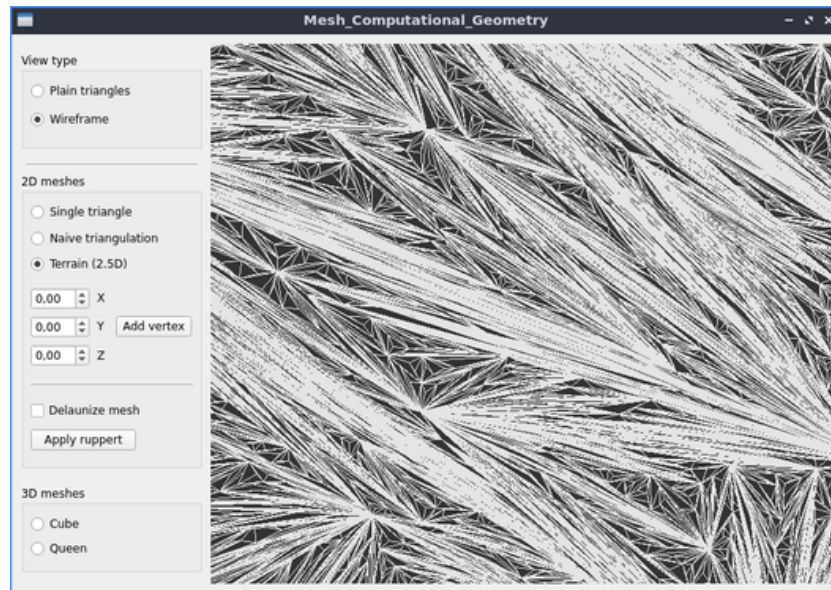
Triangle simple



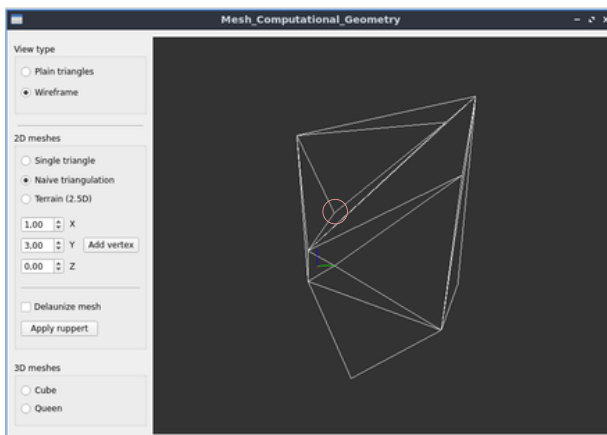
Maillage de 10 sommets aléatoires



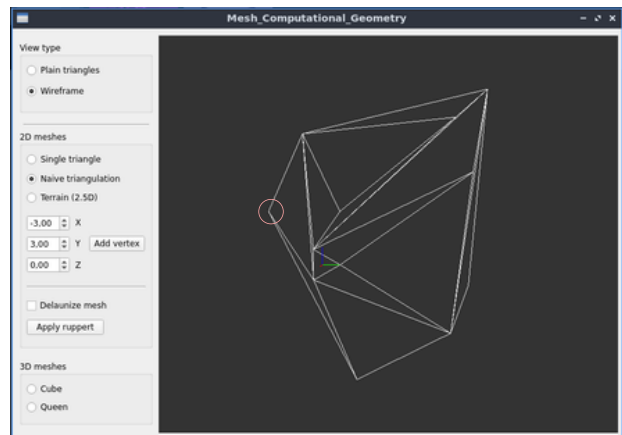
Terrain non raffiné



Ajout de sommets



Ajout du sommet aux coordonnées (1.00, 3.00, 0.00)



Ajout du sommet aux coordonnées (-3.00, 3.00, 0.00)

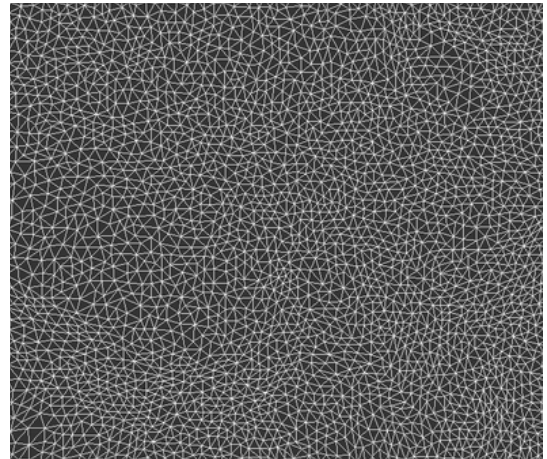
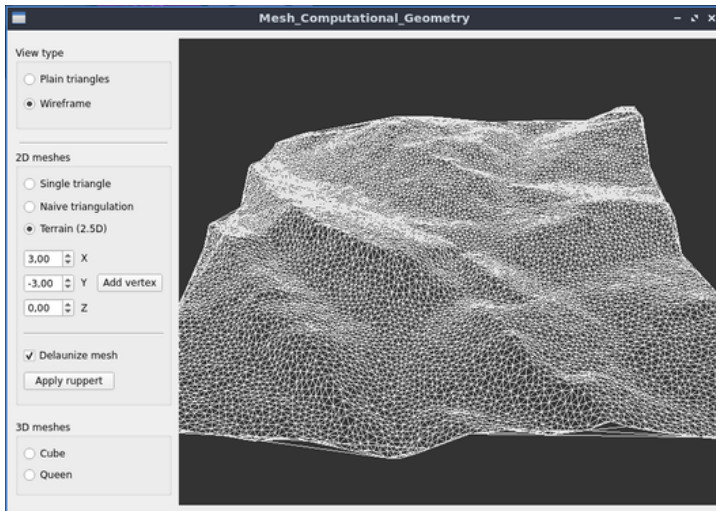
Pour l'insertion d'un sommet :

- Trouver le **triangle le plus proche**.
- Regarder si le **nouveau point est à l'intérieur du triangle** ou non.
 - S'il est à l'intérieur : on sépare le triangle en trois triangles.
 - Sinon : on ajoute deux arêtes pour former un triangle avec l'arête la plus proche. On met à jour l'enveloppe convexe.

Pour la mise à jour de l'enveloppe convexe :

- Regarder de chaque côté les **deux arêtes les plus proches** en partant du nouveau sommet, qui se situent sur l'enveloppe du maillage.
- Ajouter des **triangles incluant ces arêtes** jusqu'à ce que la convexité soit vérifiée.

Terrain raffiné

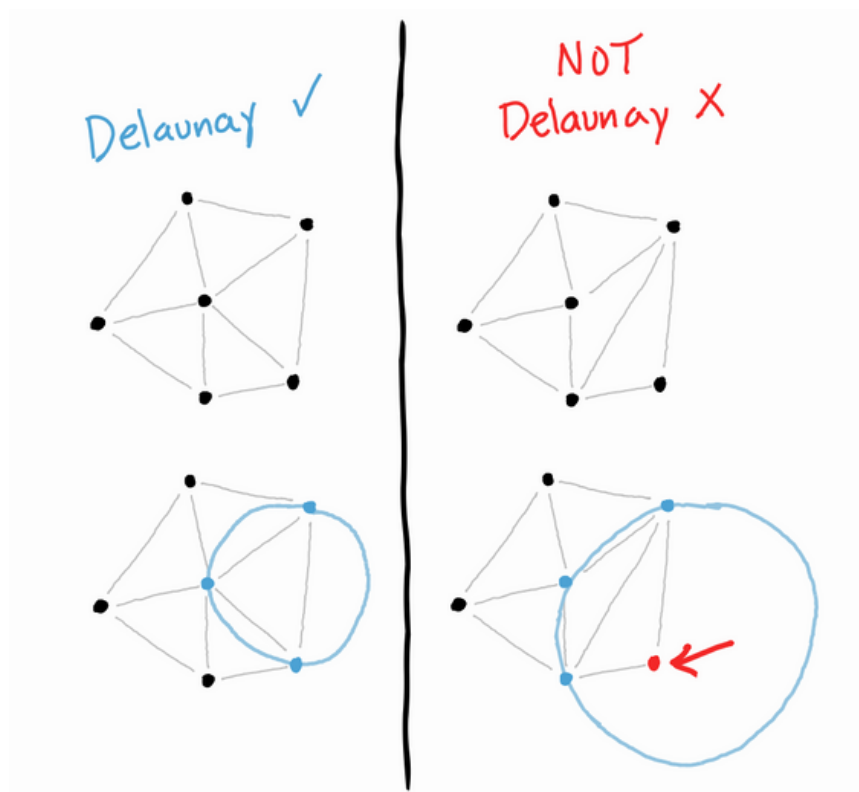


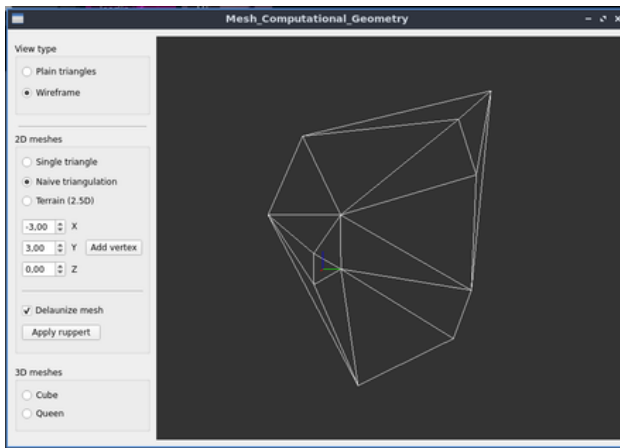
Maillage du terrain vu de haut

Maillage de 10 points raffiné

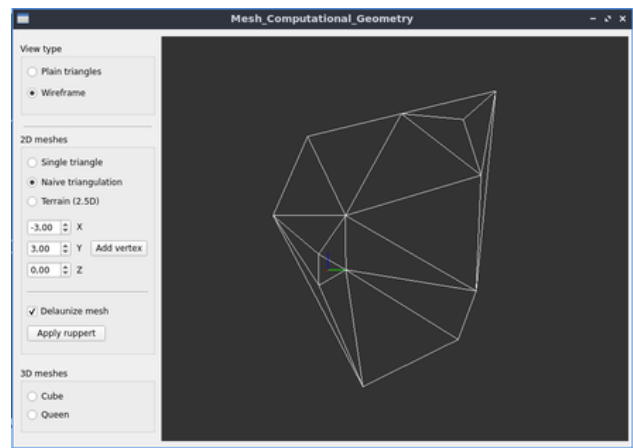
Critère de Delaunay :

- On considère deux triangles partageant une arête.
- Ayant donc deux sommets partagés et deux sommets opposés à l'arête, l'arête est de Delaunay si aucun des sommets opposés n'est dans le cercle circonscrit de l'autre triangle.





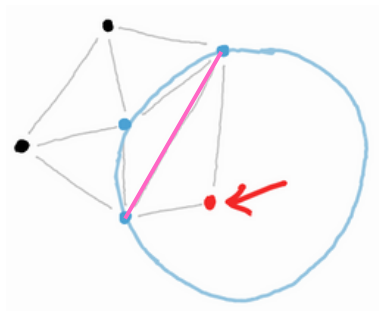
Algorithme Lawson



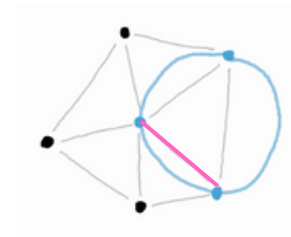
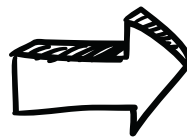
Algorithme Ruppert

Raffinement de lawson :

- On utilise une **file initialisée avec toutes les arêtes non delaunay** (test de toutes les arêtes).
- Pour chaque arête, si elle est encore non delaunay au moment de la traiter, on la **tourne** (flip) et on insère dans la file les arêtes aux alentours.
- La terminaison est garantie par la qualité du maillage strictement **croissante** au fil de l'algorithme.



Non Delaunay



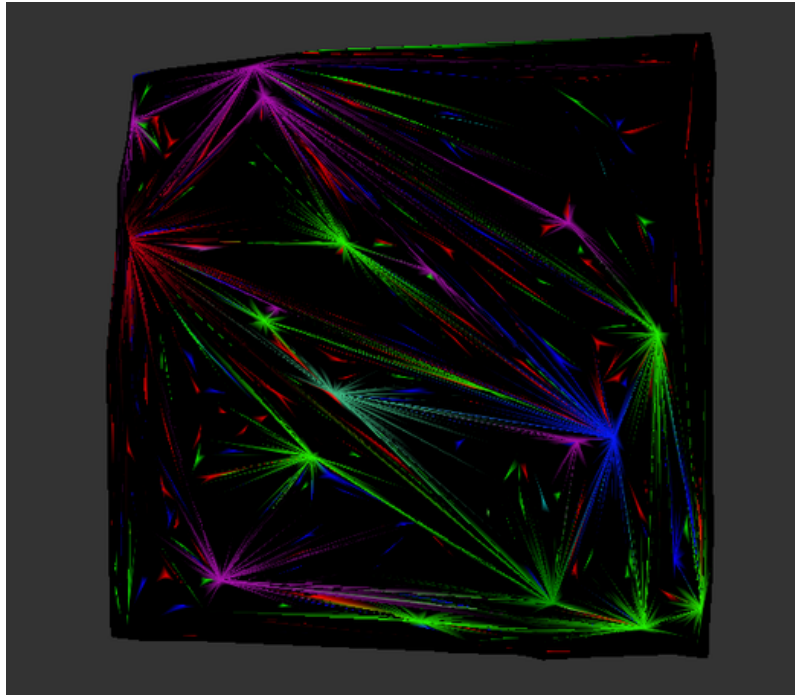
Flip de l'arête

Flip d'une arête : les triangles de chaque côté de l'arête **échanent un de leurs sommets** de façon à inverser le sens de l'arête.

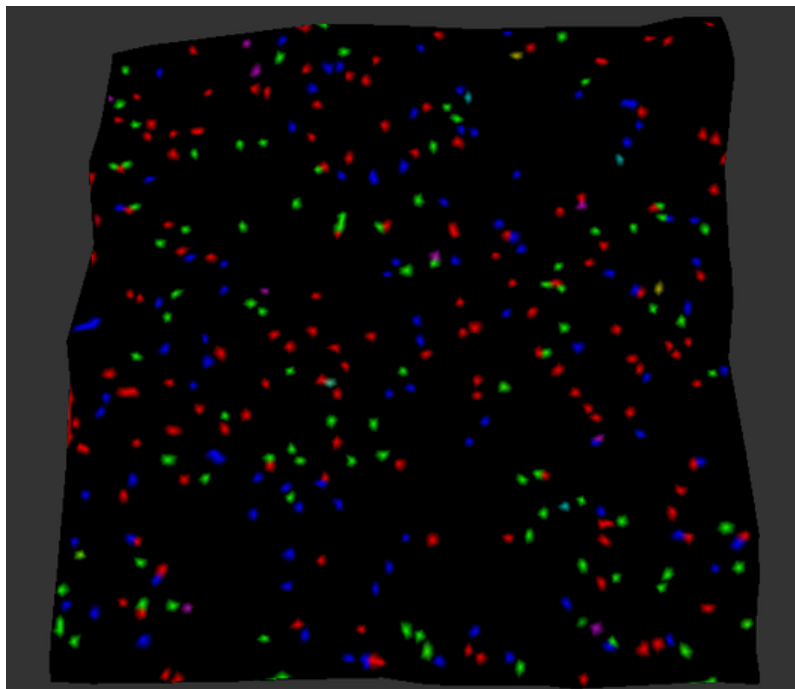
Raffinement de Ruppert (que sur un maillage de delaunay) :

- On initialise un tableau avec tous les triangles ayant un **rapport de 1:4** entre la plus petite et la plus longue arête.
- On **coupe la plus longue arête en 2**.

Laplacien du terrain (avant le raffinement)



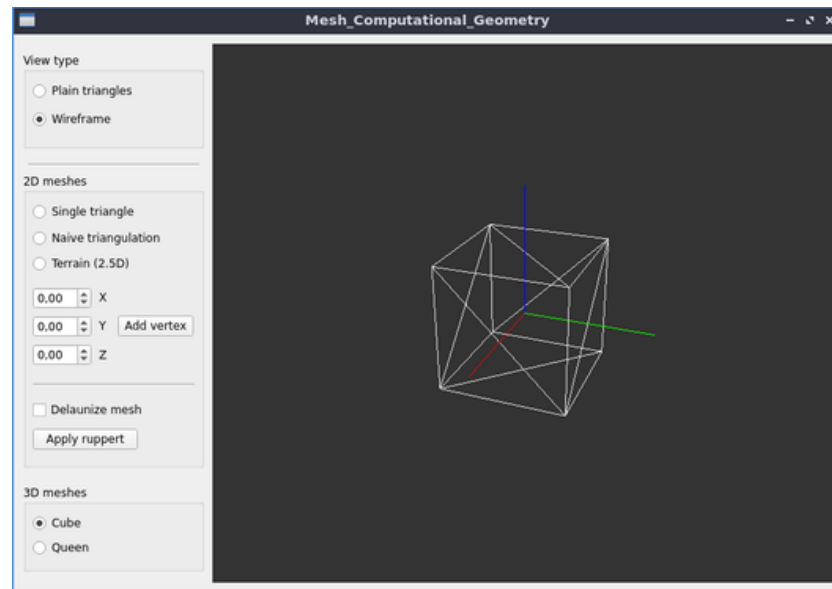
Laplacien du terrain (raffiné)



Nous avons tenté d'implémenter le *laplacien* mais les résultats ne sont pas concluants.

3D MESHES

Cube



Queen

