Compte-rendu - séance 3

LU3M101 - Projet - Maillage
Génération automatique de maillages

Licence de Sciences et Technologies

Bi-Disciplinaire Informatique et Mathématiques

Année universitaire 2019 - 2020

Encadrant : Pr Bertrand Thierry

Groupe 2 : Mlle Emilie Biegas
Mlle Zitong Yang
Mlle Alix Zheng
Mr Vincent Fu

Date de la séance : 03/03/2020

I. Récapitulatif des travaux effectués

La présentation des travaux pour cette séance a été dirigée par Alix Zheng.

Voici les points abordés :

- **Génération de nuage de points :**
  La génération des coordonnées de points s'effectue à l'aide de la librairie `random`.
  Afin d'obtenir des points suffisamment éloignés entre eux, il suffit de se fixer un seuil de distance d et de générer les points qui respectent ce seuil. Cependant, il est possible qu'un utilisateur naïf exécute notre programme sans que ce dernier se termine.
  Il est alors important de choisir des arguments cohérents pour l'exécution du programme.

- **Critère de Delaunay :**
  Cette propriété est la base du construction d'un maillage de Delaunay :
  Une triangulation est dite de Delaunay si tout cercle circonscrit à un triangle de la triangulation ne contient aucun sommet en son intérieur.
  Le critère de Delaunay a pour but de former des triangles ayant des formes convenables, autrement dit d'éviter de former si possible des triangles trop plat en deux dimensions.

- **Algorithme naïf de maillage :**
  A l'aide de l'équation de la médiatrice permettant de calculer le centre d'un cercle circonscrit d'un triangle et ainsi que des fonctions auxiliaires (calcul de distance euclidienne, calcul de toutes les combinaisons possibles de triangles (d'indices)), l'algorithme est capable de former une triangulation de Delaunay en effectuant une construction par élimination : on construit tous les triangles possibles puis on élimine tous les triangles qui ne respectent pas le critère de Delaunay.
  Quid ? Qu'en est-il des restes des triangles ? Comment sait-on que le maillage renvoyé par l'algorithme ne contient pas de "trou" ?
  En effet, la comparaison visuelle entre l'algorithme et la fonction `triplot` n'est pas suffisante pour démontrer rigoureusement la validité de l'algorithme. Ce problème est donc à vérifier si possible.
  On peut s'aider des fonctions de plot pour tracer de couleurs différentes les triangles conservés et les triangles éliminés afin de mieux discerner des cas incohérents.

- **Etude de complexité et vérification théorique :**
  Sachant qu'il y a $\binom{n}{3}$ combinaisons possibles de triangles dans un nuage à n points et que pour chaque triangle, il faut tester si tous les autres points sont dans le triangle ou pas, la complexité temporelle est donc en $O(n^4)$.
  Afin de vérifier sa complexité, le traçage de la courbe correspondant aux temps d'exécution avec la fonction `timeit` en fonction du nombre de points n dans le nuage a été effectué.
  Cependant, afin de mieux visualiser l'exposant de la complexité, il est préférable de tracer cette courbe en logarithme en fonction de $log(n)$.

- **Existence et unicité :**
  Voici quelques observations intéressantes :
  — Il n'existe pas de triangulation de Delaunay pour des points alignés.
  — Il n'y a pas unicité d'un triangle de Delaunay lorsque 4 points forment un rectangle.
  En considérant ces observations, l'algorithme va tracer les deux cas de triangulation pour 4 points formant un rectangle. Cela fait intervenir un point qui n'est pas considéré dans le nuage de points. Il faut donc essayer de régler ce problème en choisissant qu'un seul triangulation.
  Les recherches concernant les preuves de l'existence et unicité ont été difficiles.

II. Interaction pendant la séance

Voici les observations qui ont été faites pendant la séance (sachant que les remarques citées plus haut sont déjà incluses) :

- Pour le calcul de temps d'exécution, il est préférable de vérifier si la fonction `timeit` calcule bien le temps CPU et non le temps réel afin de ne pas prendre en compte en temps de calcul des autres opérations n'ayant aucun lien avec le programme.
- Concernant la preuve de l'existence et l'unicité de la triangulation de Delaunay, il faut essayer de chercher une preuve par construction inductive, autrement dit de prouver que l'ensemble des triangulations locales vérifiant le critère de Delaunay donne un triangulation global de Delaunay.

Voici quelques remarques d'ordre pratiques notamment sur l'utilisation de LaTex :

- Sur l'utilisation de Beamer, l'option fragile sur la commande frame permet de régler quelques soucis de présentation de lignes de codes avec la commande listing.
- Pour faire plusieurs colonnes sur un slide, il faut s'aider de la commande column et de partitionner les colonnes en indiquant en option le ratio par rapport à la longueur horizontale du slide (avec linewidth).

III. Programme de la séance prochaine

La séance prochaine se tiendra le 17 mars.

Voici ce que nous avons prévu de préparer pour la séance prochaine :

- Avancer sur le rapport en vu de la soumission du rapport à mi-parcours le 15 mars.
- Finaliser le deuxième algorithme de la construction de triangulation de Delaunay en vu d'améliorer si possible la complexité de ce dernier par rapport à l'algorithme naïf.
- Essayer de trouver des preuves concernant la construction d'un triangulation de Delaunay en lien avec le caractère local d'un triangulation.
- S'il reste encore du temps, essayer de réfléchir sur un maillage non plus à partir d'un nuage de points mais à partir d'un contour.