- TP 3 : Triangulations d'un ensemble de points -

Le but de ce TP est de calculer la triangulation d'un ensemble de points du plan, d'abord de façon incrémentale, puis d'améliorer cette triangulation afin d'obtenir une triangulation de Delaunay.

Le langage à utiliser est laissé libre. Toutefois, des primitives d'affichage et de tri ainsi que des trames de programme sont fournies en C++ à l'adresse :

`http://www.lirmm.fr/~bessy/AlgoGeo/accueil.html`

- Triangulation incrémentale -

- Exercice 1 -

Le but de l'exercice est d'implémenter le calcul d'une triangulation d'un ensemble de points du plan par l'algorithme incrémental vu en cours. Le résultat à obtenir est illustré Figure 1.

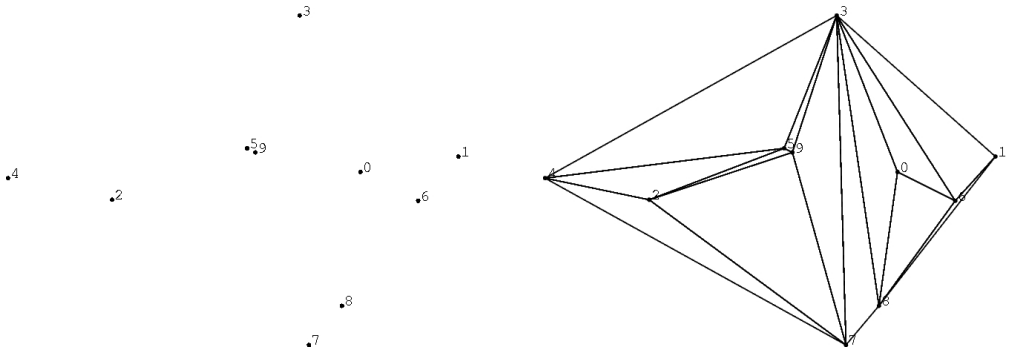

FIGURE 1 – Exemple : Un ensemble P de points du plan, puis le même ensemble triangulé et son enveloppe convexe $EC = [1, 3, 4, 7, 8, 1]$.

Dans le fichier *Triangulation.cc*, sont implémentées les fonctions et structures suivantes :

- `typedef struct {int a; int b; int c;} triangle;` : indices des sommets d'un triangle.
- `void PointAuHasard(int n, point sommet[])` : génération de sommets aléatoirement dans le plan.
- `void AffichageTriangulation(int n, point sommet[], int t, triangle T[])` : affichage des sommets et de la triangulation dont les triangles sont contenus dans T.
- `void TriLexicographique(int n, point sommet[], int t, int tri[])` : tri lexicographique des sommets, leurs indices, triés sont récupérés dans tri.

Leur usage est détaillé dans le fichier *Triangulation.cc*.

Il reste à compléter la fonction `TriangulIncrementale`, plus précisément :

1. Calcul du premier triangle et initialisation de l'enveloppe convexe (dans le sens direct !). On gère l'enveloppe convexe par un tableau dont le premier point correspond à l'indice du sommet le plus à droite ($tri[i-1]$ à l'étape i), les sommets sont ensuite énumérés dans le sens direct, et enfin on rajoute le premier sommet (voir Figure 1).
2. Puis pour l'insertion d'un nouveau point d'indice $tri[i]$:
   (a) Calcul de *kdroite*, l'indice du sommet dans l'enveloppe convexe en cours le plus loin dans le sens direct visible depuis le sommet d'indice $tri[i]$.
   (b) Calcul de *kgauche*, l'indice du sommet dans l'enveloppe convexe en cours le plus loin dans le sens indirect visible depuis le sommet d'indice $tri[i]$.
   (c) La mise-à-jour de l'enveloppe convexe.

- Triangulation de Delaunay -

- Exercice 2 -

Le but de l'exercice est d'implémenter le calcul d'une triangulation de Delaunay d'un ensemble de points à partir d'une triangulation quelconque de ces points (par exemple, celle calculée dans l'exercice 1). Le résultat à obtenir est illustré Figure 2.

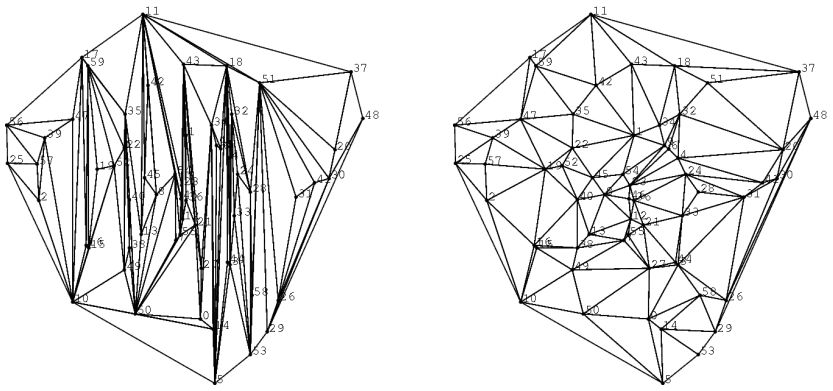

FIGURE 2 – Exemple : Un triangulation d'un ensemble de points du plan, puis sa triangulation de Delaunay.

Dans le fichier *Delaunay.cc*, se trouve une implémentation en C++ à compléter. Plus précisément :

1. Compléter la partie intitulée *Repérer un point par rapport au cercle circonscrit à un triangle.* Testez votre code avec la fonction `AffichageTestCercleCirconscrit`.
2. Compléter la fonction `Delaunay`. Il est rappelé qu'une triangulation est de Delaunay si, et seulement si, pour chaque triangle t, le cercle circonscrit à t ne contient strictement aucun point des triangles voisins à t. Les fonctions suivantes sont implémentées et leur usage est commenté dans le fichier *Delaunay.cc*.
   - `int TroisiemePoint(triangle s, triangle t)` : retourne l'indice du point du triangle s qui n'est pas dans le triangle t.
   - `void Flip(int i, int j, int t, triangle T[], int voisin[][3])` : effectue le 'flip' de l'arête commune entre les triangles $T[i]$ et $T[j]$, le tableau *voisin*[][3] est remis à jour.

- Exercices supplémentaires -

- Exercice 3 - Arbre couvrant euclidien minimum -

Calculer un arbre couvrant euclidien minimum d'un ensemble de points aléatoirement répartis (pour ceux qui ont suivi le module d'algo de graphes en L3, cela correspond au tp sur l'algo de Kruskal...). Faire afficher celui-ci en surimpression sur la triangulation de Delaunay de l'ensemble de points choisis, et vérifier que les arêtes de l'arbre sont des arêtes de la triangulation.

- Exercice 4 - Diagramme de Voronoï -

A l'aide de la triangulation de Delaunay obtenue précédemment, écrire une fonction qui calcule les points du diagramme de Voronoï.
Ecrire une seconde fonction qui finit le calcul et trace effectivement ce diagramme (attention aux triangles bordant la face externe...).
Si vous avez des loisirs, effectuer le calcul du diagramme de Voronoï directement, à l'aide de l'algorithme de S.Fortune vu en cours.