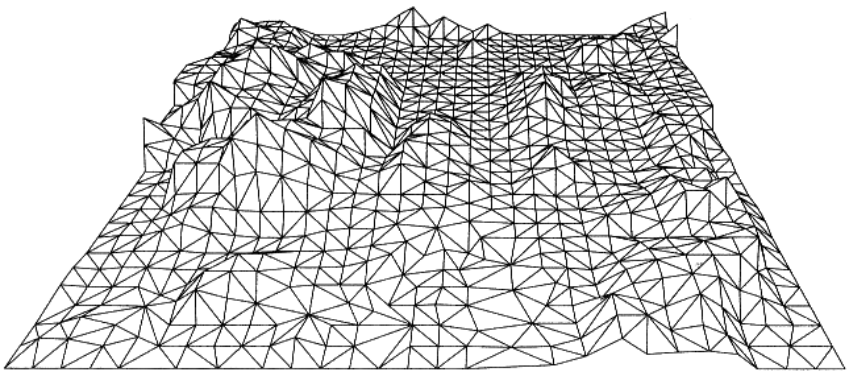
Triangulations d'un ensemble de points du plan

I. Introduction

Dans l'introduction du cours sur les enveloppes convexes, nous avons déjà évoqué le fait que les triangulations d'ensembles de points sont une des applications de la géométrie algorithmique. Considérons par exemple une modélisation d'un terrain en 3D telle qu'on peut en voir dans Google Earth.

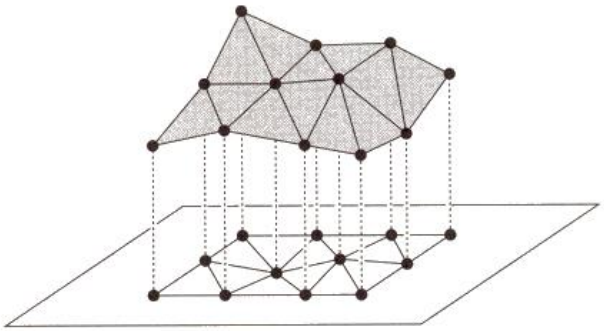
Image 3D extraite de Google Earth.

Ce modèle numérique de terrain est obtenu par la triangulation d'un ensemble de points 3D qui ont été récupérés sur le terrain réel, par exemple à l'aide d'un système laser aéroporté (communément appelé lidar).



Triangulation d'un ensemble de points représentant un terrain.
Figure extraite du livre « Computational Geometry – Algorithms and Applications »
de Mark de Berg, Otfried Cheong, Marc van Kreveled et Mark Overmars.

Dans le cas de la modélisation de terrain, une triangulation de l'ensemble de points 3D peut être construite en ignorant l'altitude des points. Ceci revient à construire une triangulation d'un ensemble de points 2D.



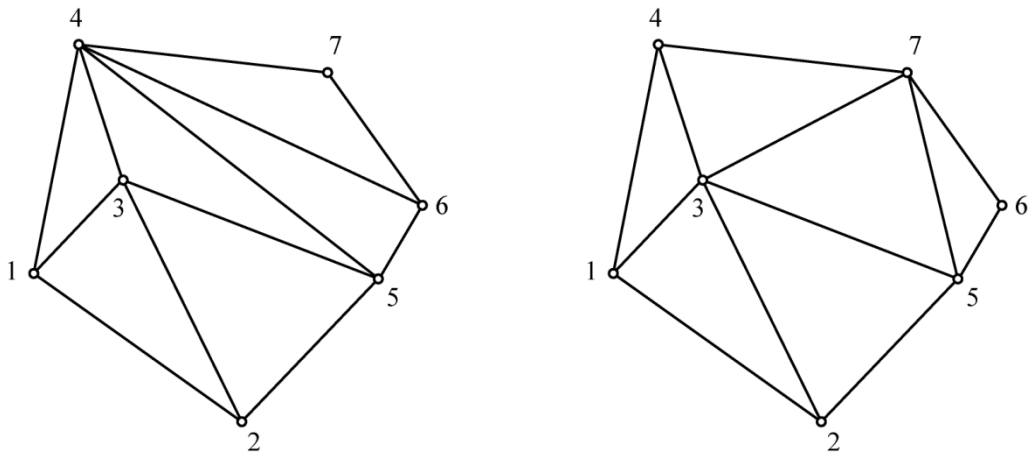
Les points 3D sont projetés dans le plan et sont triangulés.
Figure extraite du livre « Computational Geometry – Algorithms and Applications »
de Mark de Berg, Otfried Cheong, Marc van Kreveled et Mark Overmars.

Il est donc nécessaire d'avoir des algorithmes efficaces pour construire des triangulations d'ensembles de points du plan.

II. Définition et propriétés

Soit P un ensemble de n points dans le plan. On suppose, dans un premier temps, que P ne contient pas trois points alignés.

On appelle **triangulation** de P, toute partition de l'enveloppe convexe de P en triangles dont les sommets sont les points de P. Il y a en général différentes façon de trianguler un ensemble de points.



Deux triangulations d'un même ensemble de points.

Dans la plupart des applications, il est important que la triangulation construite soit la plus régulière possible, c'est-à-dire que ses triangles ressemblent le plus possible à des triangles équilatéraux. Dans l'exemple ci-dessus, on préférera clairement la triangulation de droite à la triangulation de gauche.

Une propriété très intéressante des triangulations est que, si l'ensemble de points P est fixé, alors toutes les triangulations de P ont le même nombre de triangles. Ce nombre est égal à **2n-n'-2**, où n' est le nombre de sommets de l'enveloppe convexe de P. Toutes les triangulations de P ont également le même nombre de côtés. Ce nombre est égal à **3n-n'-3**.

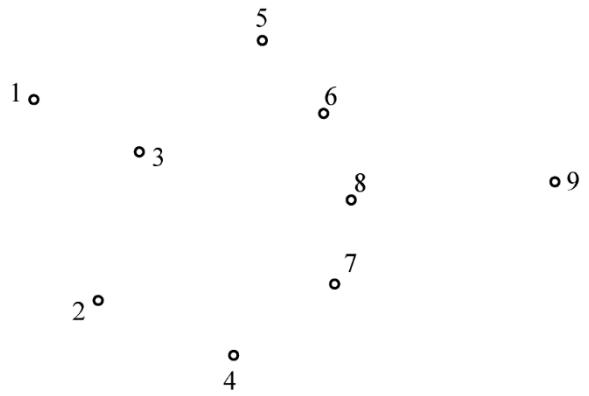
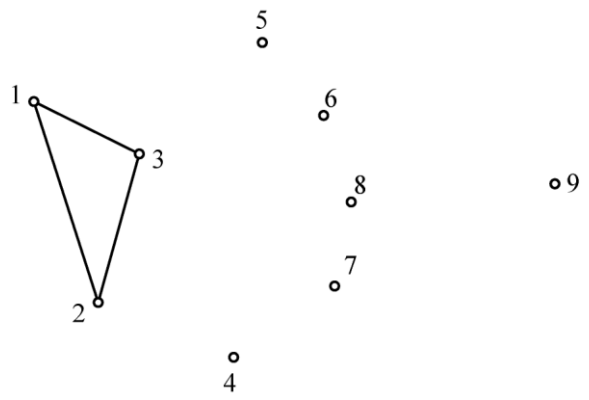
Avant d'étudier un algorithme qui permet de construire la triangulation la plus régulière possible d'un ensemble de points, nous donnons d'abord un algorithme simple qui permet de construire une première triangulation de cet ensemble de points.

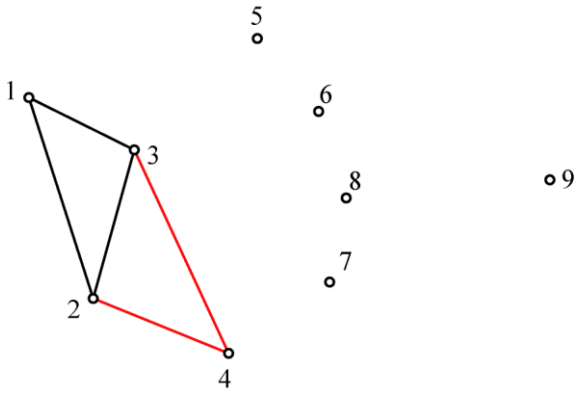
III. Construction incrémentale d'une triangulation

Une triangulation d'un ensemble de points P peut être construite facilement en adaptant l'algorithme incrémental de construction d'enveloppe convexe vu dans le premier cours. On rappelle que dans cet algorithme les points sont d'abord triés par abscisses croissantes et sont ensuite ajoutés un à un à l'enveloppe convexe courante. À chaque ajout d'un point p, l'algorithme détermine la ligne polygonale des côtés de l'enveloppe convexe courante qui doivent être supprimés, c'est-à-dire les côtés qui ont p « à leur droite ».

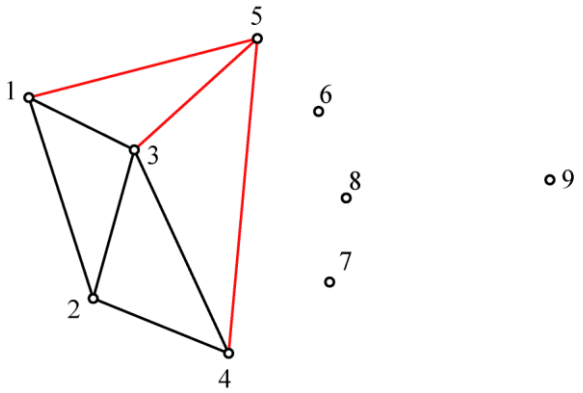
Pour construire une triangulation de P, on procède de manière similaire. Lors de l'ajout d'un nouveau point p, on détermine d'abord la ligne polygonale formée par les côtés de l'enveloppe convexe courante qui ont le point p « à leur droite ». Ensuite, au lieu de supprimer cette ligne polygonale, on relie chacun de ses sommets au point p.

Exemple de déroulement de l'algorithme

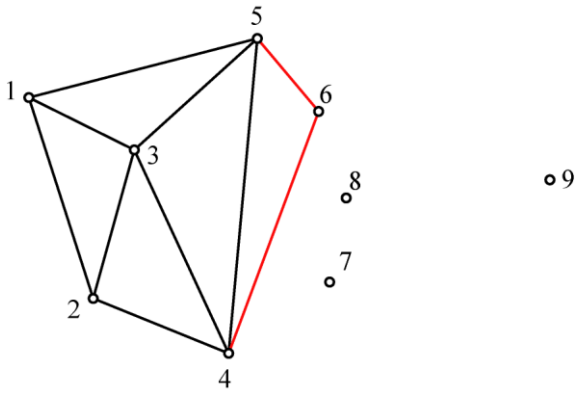
<table>
<tr>
<td> 
</td>
<td>Pour construire une triangulation de cet ensemble de 9 points, l'algorithme commence par trier les points par abscisses croissantes.</td>
</tr>
<tr>
<td> 
</td>
<td>Il construit ensuite le triangle formé par les 3 premiers points. On obtient ainsi une triangulation de ces 3 points.</td>
</tr>
</table>



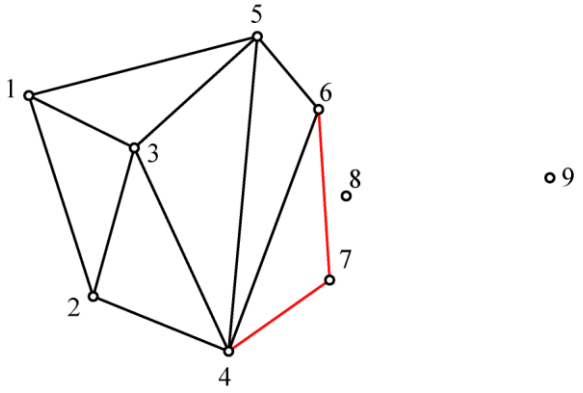
Lorsque le point 4 est ajouté, seul le côté 23 a le point 4 « à sa droite ». Le point 4 est alors relié aux points 2 et 3. On obtient ainsi une triangulation des 4 premiers points.



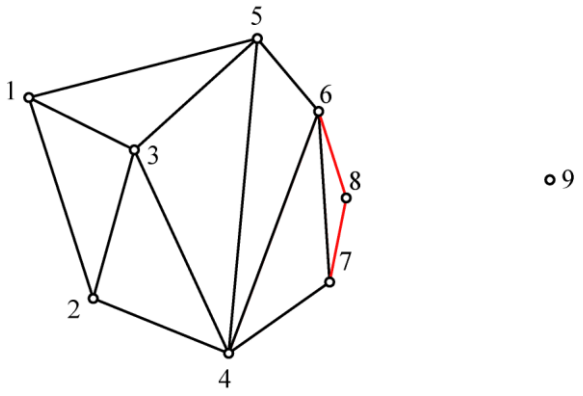
Lorsque le point 5 est ajouté, les côtés 43 et 31 sont les côtés de l'enveloppe convexe courante qui ont le point 5 « à leur droite ». Le point 5 est alors relié aux points 4, 3 et 1. Notons que cela crée deux nouveaux triangles.



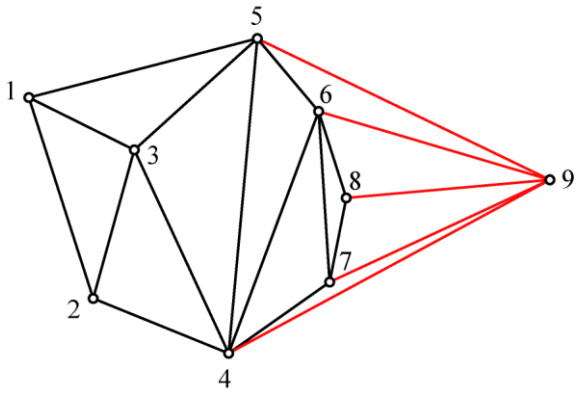
Le point 6 est ensuite relié à 4 et à 5. Cette opération ajoute dans la triangulation un sommet, deux côtés et un triangle.



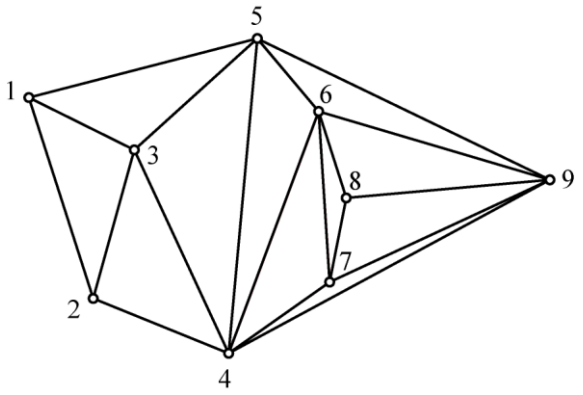
Puis le point 7 est relié à 4 et à 6.



Et le point 8 est relié à 7 et à 6.



Lors de l'ajout du point 9, les côtés 47, 78, 86 et 65 sont les côtés de l'enveloppe convexe courante qui ont 9 « à leur droite ». Les sommets de la ligne polygonale formée par ces côtés sont 4, 7, 8, 6 et 5. Ils sont tous reliés au point 9.



On obtient ainsi une triangulation de cet ensemble de 9 points. Cette triangulation comporte 19 côtés et 11 triangles.

IV. Complexité de l'algorithme incrémental

L'algorithme comporte trois phases principales :

1. le tri des points,
2. la construction d'un triangle avec les 3 premiers points,
3. l'ajout des n-3 autres points un à un.

Les meilleurs algorithmes de tri permettent de trier un ensemble de n valeurs en temps O(n log n). La complexité de la première phase de l'algorithme est donc en O(n log n).

La construction du premier triangle peut se faire en temps constant. La deuxième phase de l'algorithme a donc une complexité en O(1).

Lors de la troisième phase de l'algorithme, à chaque ajout d'un nouveau point p, l'algorithme recherche les côtés de l'enveloppe convexe courante qui « ont le point p à leur droite » et relie les extrémités de ces côtés au point p. La complexité de l'ajout d'un point p dépend donc du nombre de côtés qui sont parcourus lors de cet ajout. Afin de réduire cette complexité, il faut parcourir le moins de côtés possibles. L'idée est d'utiliser la même méthode que dans l'algorithme incrémental de construction de l'enveloppe convexe :

- on part du sommet d'abscisse maximale de l'enveloppe convexe courante et on tourne dans le sens trigonométrique jusqu'à trouver un côté qui n'a pas p « à sa droite »,
- on repart du sommet d'abscisse maximale de l'enveloppe convexe courante et on tourne dans l'autre sens jusqu'à trouver un côté qui n'a pas p « à sa droite ».

Cette méthode parcourt tous les côtés qui « ont p à leur droite » et ne parcourt que deux côtés qui « ont p à leur gauche » (les deux côtés sur lesquels les deux boucles de recherche s'arrêtent). Or, lors de la phase 3, n-3 points sont ajoutés en tout. Le nombre **total** de côtés parcourus pendant toute la phase 3 et qui « ont le point ajouté à leur gauche » est donc égal à **2(n-3)**.

Il reste à compter le nombre de côtés parcourus lors de la phase 3 et qui « ont le point ajouté à leur droite ». Notons que les extrémités de tous ces côtés sont reliées au point ajouté, créant ainsi un triangle par côté. De plus, tous les triangles de la triangulation, à part le premier, sont construits de cette manière. Or, on sait que notre triangulation contient 2n-n'-2 triangles (où n' est le nombre de sommets de l'enveloppe convexe). Il en résulte que le nombre **total** de côtés parcourus par la phase 3 et qui « ont le point ajouté à leur droite » est égal à **2n-n'-3**.

Le nombre total de côtés parcourus lors de la phase 3 de l'algorithme est donc égal à 2(n-3) + 2n-n'-3 = 4n-n'-9. Il en résulte que la complexité de la troisième phase est en O(n).

La phase dont la complexité est la plus grande est la première phase. Il en résulte que la complexité totale de l'algorithme de triangulation est égale à la complexité du tri, c'est à dire en O(n log n).

V. Implémentation de l'algorithme incrémental

Exercice 1 :
Écrire la fonction `void triangulation(vector<Point> &T, Carte &C)` qui, étant donné un tableau de points `T`, construit dans la carte combinatoire `C` une triangulation de cet ensemble de points avec l'algorithme ci-dessus. On conservera dans le demi-côté particulier de `C` un demi-côté de la face externe de `C`.

Aides :
- il est plus simple, dans la phase 2 de l'algorithme, de ne traiter que deux points (à la place de trois),
- les côtés de l'enveloppe convexe sont les côtés de la face externe de la carte,
- le point d'abscisse maximale de l'enveloppe convexe courante est toujours relié au point ajouté.

Exercice 2 :
Modifier la fonction précédente de telle sorte qu'elle fonctionne également dans le cas où l'ensemble de points peut contenir plus de deux points alignés.