

# Triangulation de Delaunay

Massicot Pillet

Ecole Normale Supérieure de Cachan, antenne de Bretagne

avril 2010

- 1 Introduction
- 2 Analyse Mathématique
- 3 Algorithmes
- 4 Correction et Complexité
- 5 Conclusion

## Définition (triangulation)

On appelle triangulation d'un ensemble  $P$  de points du plan, la donnée d'un ensemble de triangles  $\mathcal{T}$  dont les sommets sont des points de  $P$  tel que deux triangles distincts ont leurs intérieurs d'intersection vide, et qui maximise le nombre d'arêtes.

## Définition (triangulation)

On appelle triangulation d'un ensemble  $P$  de points du plan, la donnée d'un ensemble de triangles  $\mathcal{T}$  dont les sommets sont des points de  $P$  tel que deux triangles distincts ont leurs intérieurs d'intersection vide, et qui maximise le nombre d'arêtes.

## Définition (triangulation de Delaunay)

Une triangulation  $\mathcal{DT}$  d'un ensemble  $P$  de points du plan est dite de Delaunay si l'intérieur de tout triangle de  $\mathcal{DT}$  ne contient aucun point de  $P$ .

## Définition (triangulation)

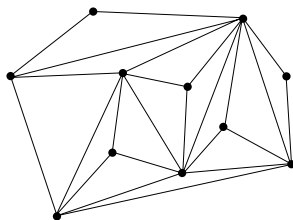
On appelle triangulation d'un ensemble  $P$  de points du plan, la donnée d'un ensemble de triangles  $\mathcal{T}$  dont les sommets sont des points de  $P$  tel que deux triangles distincts ont leurs intérieurs d'intersection vide, et qui maximise le nombre d'arêtes.

## Définition (triangulation de Delaunay)

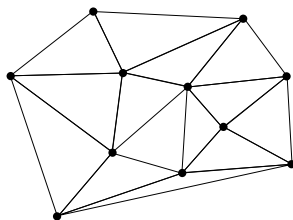
Une triangulation  $\mathcal{DT}$  d'un ensemble  $P$  de points du plan est dite de Delaunay si l'intérieur de tout triangle de  $\mathcal{DT}$  ne contient aucun point de  $P$ .

## Remarque

*La triangulation de Delaunay est unique.*



(a)

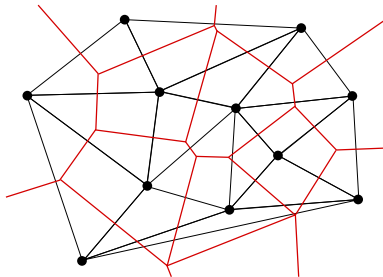


(b)

Exemples de deux triangulations d'un ensemble de points du plan : **(a)** n'est pas de Delaunay alors que **(b)** l'est.

## Définition

Soit  $P$  un ensemble de points du plan, on appelle cellule de Voronoï du germe  $s \in P$  l'ensemble des points du plan plus proche de  $s$  que de n'importe quel autre point de  $P$ . On appelle diagramme de Voronoï la frontière de tous les cellules. L'intérieur des cellules et le diagramme forment une partition du plan.







- Calcul d'un arbre couvrant minimal

- Calcul d'un arbre couvrant minimal
- Maillage

- Calcul d'un arbre couvrant minimal
- Maillage
- Parcours en champ miné

## 1 Introduction

## 2 Analyse Mathématique

- Triangulation d'un quadrilatère
- Théorème d'Euler
- Unicité de la triangulation de Delaunay
- Maximisation du minimum des angles
- Condition locale de Delaunay

## 3 Algorithmes

## 4 Correction et Complexité

## 5 Conclusion

## Lemme

*Soit  $ABCD$  un quadrilatère convexe non dégénéré du plan. Alors il possède exactement 2 triangulations dont l'une au moins est de Delaunay*

## Lemme

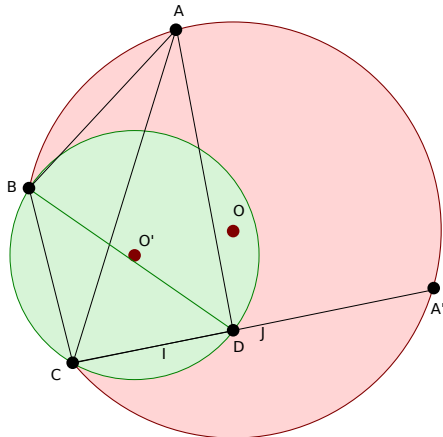
*Soit  $ABCD$  un quadrilatère convexe non dégénéré du plan. Alors il possède exactement 2 triangulations dont l'une au moins est de Delaunay*

## Démonstration

les deux triangulations possibles sont  $\{ABC, ACD\}$  et  $\{ABD, BCD\}$ .  
Supposons que la première ne soit pas de Delaunay. L'un des deux triangles contient donc le quatrième point à l'intérieur de son cercle circonscrit. Quitte à renommer les sommets, on peut considérer que  $D \in \mathcal{D}_{ABC}$ .

Construisons le point  $A'$  intersection du cercle  $\mathcal{C}_{ABC}$  avec la droite  $(CD)$ , par hypothèse comme  $D \in \mathcal{D}_{ABC}$  on sait que  $d(C, D) < d(C, A') = d(C, A)$ .

La droite  $\Delta$  médiatrice du segment  $[A', C]$  sépare le plan en deux demi-plan qui représentent respectivement l'ensembles des points plus proches de  $C$  et l'ensemble des points plus proches de  $A'$ .



## Démonstration (suite)

Soit  $L$  la médiatrice du segment  $[D, C]$ , par hypothèse elle est parallèle et non confondue avec  $\Delta$ . Ainsi elle est dans l'un des deux demi-plans : nécessairement celui des points plus proches de  $C$  car elle contient le milieu du segment  $[D, C]$  qui est plus proche de  $C$  que le milieu du segment  $[A', C]$  qui lui est sur  $\Delta$ .

Soit donc  $O'$  l'intersection de  $L$  avec la médiatrice du segment  $[B, C]$ , c'est le centre du cercle circonscrit à  $BCD$ , et on sait que  $d(O', C) < d(O, C)$

Par suite, comme le rayon du cercle circonscrit à  $BCD$  est strictement plus petit que celui passant par  $ABC$ ,  $A \notin \mathcal{D}_{BCD}$



## Démonstration (suite)

Soit  $L$  la médiatrice du segment  $[D, C]$ , par hypothèse elle est parallèle et non confondue avec  $\Delta$ . Ainsi elle est dans l'un des deux demi-plans : nécessairement celui des points plus proches de  $C$  car elle contient le milieu du segment  $[D, C]$  qui est plus proche de  $C$  que le milieu du segment  $[A', C]$  qui lui est sur  $\Delta$ .

Soit donc  $O'$  l'intersection de  $L$  avec la médiatrice du segment  $[B, C]$ , c'est le centre du cercle circonscrit à  $BCD$ , et on sait que  $d(O', C) < d(O, C)$

Par suite, comme le rayon du cercle circonscrit à  $BCD$  est strictement plus petit que celui passant par  $ABC$ ,  $A \notin \mathcal{D}_{BCD}$

De plus  $C \notin \mathcal{D}_{ABD} \Leftrightarrow A \notin \mathcal{D}_{BCD}$ , en effet, les centres des deux cercles circonscrits sont sur la médiatrice de  $[BD]$ . Dans le cas limite ils sont confondus, et quand  $C \in \mathcal{D}_{ABD}$ , alors le centre de  $\mathcal{C}_{BCD}$  est plus loin de  $[BD]$  que l'autre centre, le disque contient donc  $A$ .

## Théorème (Euler)

*Soit  $\mathcal{T}$  une triangulation d'un ensemble de points  $P$ . Soient  $t$  son nombre de triangles,  $a$  son nombre d'arêtes et  $n = |P|$  son nombre de sommets. Alors ces trois grandeurs sont liées par la relation :  $t - a + n = 1$*

## Théorème (Euler)

*Soit  $\mathcal{T}$  une triangulation d'un ensemble de points  $P$ . Soient  $t$  son nombre de triangles,  $a$  son nombre d'arêtes et  $n = |P|$  son nombre de sommets. Alors ces trois grandeurs sont liées par la relation :  $t - a + n = 1$*

## Démonstration.

On peut démontrer le résultat par récurrence descendante :

On enlève un sommet de l'enveloppe convexe, en supprimant ainsi, les  $k$  triangles qu'il formait avec les autres points de  $P$  et les  $k + 1$  arêtes qui délimitaient ces triangles.

On a toujours :  $t - a + n = (t - k) - (a - (k + 1)) + (n - 1)$ . Donc la grandeur appelée caractéristique d'Euler-Poincaré est conservée.

C'est donc la même que celle dans le cas du triangle seul :

$$1 - 3 + 3 = 1.$$



## Corollaire

*Soit  $P$  un ensemble de  $n$  points du plan. On note  $q$  le nombre d'arêtes de son enveloppe convexe. Alors si  $\mathcal{T}$  est une triangulation de  $P$  :*

- *son nombre de triangles  $t = 2n - 2 - q$*
- *son nombre d'arêtes  $a = 3n - 3 - q$*

## Corollaire

Soit  $P$  un ensemble de  $n$  points du plan. On note  $q$  le nombre d'arêtes de son enveloppe convexe. Alors si  $\mathcal{T}$  est une triangulation de  $P$  :

- son nombre de triangles  $t = 2n - 2 - q$
- son nombre d'arêtes  $a = 3n - 3 - q$

## Corollaire

Soit  $\mathcal{DT}$  une triangulation de Delaunay, d'un ensemble  $P$  du plan. Alors, en moyenne, le nombre de voisins d'un sommet  $p \in P$  est inférieur à 6.

## Corollaire

*Soit  $P$  un ensemble de  $n$  points du plan. On note  $q$  le nombre d'arêtes de son enveloppe convexe. Alors si  $\mathcal{T}$  est une triangulation de  $P$  :*

- *son nombre de triangles  $t = 2n - 2 - q$*
- *son nombre d'arêtes  $a = 3n - 3 - q$*

## Corollaire

*Soit  $\mathcal{DT}$  une triangulation de Delaunay, d'un ensemble  $P$  du plan. Alors, en moyenne, le nombre de voisins d'un sommet  $p \in P$  est inférieur à 6.*

Ce nombre sera noté  $k$  par la suite, il est considéré comme une variable aléatoire dépendant de la variable aléatoire  $p$ .

## Théorème

*Soit un ensemble  $P$  de points du plan (en position générale) alors il existe une unique triangulation de  $P$  qui soit de Delaunay.*

## Théorème

*Soit un ensemble  $P$  de points du plan (en position générale) alors il existe une unique triangulation de  $P$  qui soit de Delaunay.*

## Démonstration.

L'existence d'une telle triangulation sera démontrée par les algorithmes que nous allons vous proposer.

L'unicité vient de l'unicité du diagramme de Voronoï





## Théorème

*Soit  $P$  un ensemble de points du plan, la triangulation de Delaunay, est la triangulation qui maximise l'ordre lexicographique sur les angles*

## Théorème

*Soit  $P$  un ensemble de points du plan, la triangulation de Delaunay, est la triangulation qui maximise l'ordre lexicographique sur les angles*

## Corollaire

*La triangulation de Delaunay maximise le plus petit des angles.*

## Définition

On appelle voisin d'un triangle d'une triangulation  $\mathcal{T}$ , les triangles qui partagent au moins une arête avec celui-ci.

On dit qu'une triangulation  $\mathcal{T}$  est localement de Delaunay en  $T \in \mathcal{T}$  si  $\mathcal{D}_T$  n'intersecte aucun sommet des voisins de  $T$  qui ne soient déjà dans  $T$ .

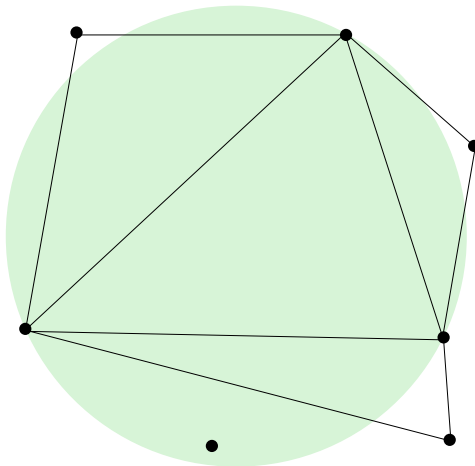
## Définition

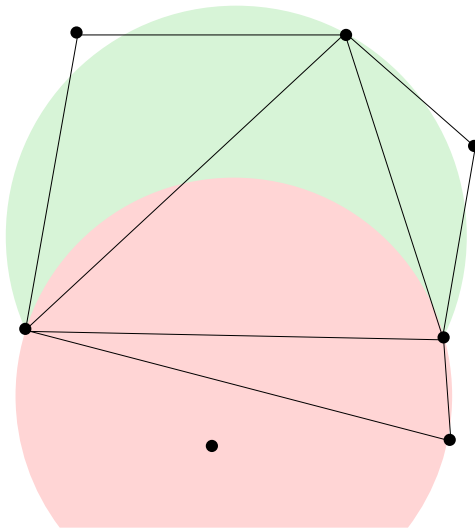
On appelle voisin d'un triangle d'une triangulation  $\mathcal{T}$ , les triangles qui partagent au moins une arête avec celui-ci.

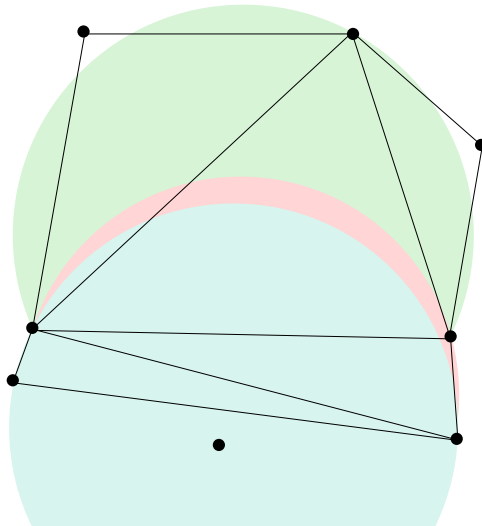
On dit qu'une triangulation  $\mathcal{T}$  est localement de Delaunay en  $T \in \mathcal{T}$  si  $\mathcal{D}_T$  n'intersecte aucun sommet des voisins de  $T$  qui ne soient déjà dans  $T$ .

## Théorème

*Une triangulation  $\mathcal{T}$  d'un ensemble  $P$  est localement de Delaunay partout ssi elle est de Delaunay*







- 1 Introduction
- 2 Analyse Mathématique
- 3 Algorithmes
  - Algorithme (théorique) par basculement
  - Structure de données
    - D.A.G.
    - Triangle
  - Algorithme incrémental
- 4 Correction et Complexité
- 5 Conclusion



---

## Algorithme 1 : Algorithme par basculements

---

**Entrées** :  $P$  ensemble de points du plan

**Sorties** :  $\mathcal{T}$  triangulation de Delaunay de l'ensemble  $P$

**début**

    Construire une triangulation  $\mathcal{T}$

**tant que**  $\mathcal{T}$  *n'est pas de Delaunay* **faire**

        Trouver un quadrilatère  $Q$  tel que la triangulation de  $Q$  ne soit pas de Delaunay.

        Changer la triangulation de  $Q$ .

**renvoyer**  $\mathcal{T}$

**fin**

---

On note  $k$  le nombre de triangles incompatibles avec un point  $x$ .

On classe les triangles de  $\mathcal{T}$  sous forme d'un graphe orienté sans cycles (un arbre avec plusieurs pères possibles) que l'on construit au cours de l'algorithme.

A chaque ajout de point, le nombre de fils d'un nœud est majoré par  $k$ .  
Les procédures liées à cette structure sont :

```
NouveauNoeud([Père1,Père2,...],triangle)
Fils(noeud)
Pères(noeud)
Supprime(noeud)
```

Pour optimiser le parcours du DAG "vers le bas", les nœuds seront stockés en mémoire sous la forme d'un triangle, puis de la donnée de ses fils.

On considérera la structure Triangle qui consistera en la donnée de trois points  $s_1, s_2, s_3$  (sommets) et de trois pointeurs  $T_1, T_2, T_3$  vers les éventuels triangles voisins par une arête (avec la convention  $T_i$  triangle opposé au sommet  $s_i$ ).

On ajoutera éventuellement, un bit de marquage pour le parcours en pile. Mais par abus de notation, on désignera un triangle par la seule donnée du triplet de ses sommets.

---

## Algorithme 2 : Algorithme incrémental simplifié

---

**Entrées** :  $P$  ensemble de points du plan

**Sorties** :  $\mathcal{T}$  triangulation de Delaunay de l'ensemble  $P$

**début**

Considérer  $u, v, w$  trois point du plan tels que le triangle  $(uvw)$  contienne  $P$

$\mathcal{T} \leftarrow \{(uvw)\}$

NouveauNoeud ([Nul],  $(u, v, w)$ )

**tant que**  $P \neq \emptyset$  **faire**

    Enlever  $x$  de  $P$

$N \leftarrow \text{TrouverTriangleContenant}(x)$

    RearrangerTriangulation( $x, N$ )

Supprimer  $u, v, w$  et les triangles qui en partent.

**renvoyer**  $\mathcal{T}$

**fin**

---

Parcours du DAG : On cherche le triangle de la triangulation déjà construite qui contient  $x$ .

---

**Fonction** TrouverTriangleContenant( $x$ )

---

$N \leftarrow (uvw)$

**tant que** Fils( $N$ )  $\neq \emptyset$  **faire**

$M \leftarrow \text{FilsAiné}(N)$

**si**  $x \in M$  **alors**

$N \leftarrow M$

**sinon**

$M \leftarrow \text{Frère}(N)$

**renvoyer**  $N$

---

---

## Fonction `RearrangerTriangulation(x, N)`

---

Empiler  $N$

**tant que**  $PILE \neq \emptyset$  **faire**

    lire  $L$  sur le sommet de la pile

**si**  $M$  voisin de  $L$  non pointé **alors**

        Pointer ( $M$ )

**si**  $x \in \mathcal{D}_M$  **alors**

            Empiler ( $M$ )

**sinon**

            Soient  $u, v$  les sommets de  $M \cap L$

            Constuire le triangle ( $uvx$ )

            NouveauNoeud ( $PILE, (u, v, x)$ )

**sinon**

        Depiler  $L$

Supprimer le triangle  $L$

---

- 1 Introduction
- 2 Analyse Mathématique
- 3 Algorithmes
- 4 Correction et Complexité**
  - Correction
  - Borne inférieure
  - Cas pire et moyenne
- 5 Conclusion

Il est clair qu'il suffit de montrer que la procédure `RearrangerTriangulation(x,N)` va transformer une triangulation de Delaunay, en une nouvelle triangulation de Delaunay.

On appellera  $E$  l'ensemble des triangles en conflit avec  $x$



Il est clair qu'il suffit de montrer que la procédure `RearrangerTriangulation(x,N)` va transformer une triangulation de Delaunay, en une nouvelle triangulation de Delaunay.

On appellera  $E$  l'ensemble des triangles en conflit avec  $x$

Pour cela, on va montrer que la nouvelle triangulation est localement de Delaunay en tout triangle.

Il est clair qu'il suffit de montrer que la procédure `RearrangerTriangulation(x,N)` va transformer une triangulation de Delaunay, en une nouvelle triangulation de Delaunay.

On appellera  $E$  l'ensemble des triangles en conflit avec  $x$

Pour cela, on va montrer que la nouvelle triangulation est localement de Delaunay en tout triangle.

En effet : pour tout triangle extérieur à  $E$ , ils ne sont par définition pas en conflit avec  $x$  et donc comme la triangulation était précédemment de Delaunay, ils sont tous localement de Delaunay.

Pour tout nouveau triangle, il est clair que leur cercle circonscrit ne contient aucun autre sommet de  $E$ , en effet :

- Si  $(abc)$  était dans  $E$ , alors le quadrilatère  $(abxc)$  (ou  $(abcx)$ ) ne pourrait se trianguler en  $(abc)$  et  $(acx)$  car  $(abc)$  est en conflit avec  $x$ .
- Si  $d$  est un voisin de  $a$  (ou  $b$ ) dans les sommets de  $E$ , et qu'il est en conflit avec  $(abx)$ , alors il était nécessairement en conflit avec  $(abc)$ , car  $x$  est en conflit avec  $(abc)$  par hypothèse.

Pour tout nouveau triangle, il est clair que leur cercle circonscrit ne contient aucun autre sommet de  $E$ , en effet :

- Si  $(abc)$  était dans  $E$ , alors le quadrilatère  $(abxc)$  (ou  $(abcx)$ ) ne pourrait se trianguler en  $(abc)$  et  $(acx)$  car  $(abc)$  est en conflit avec  $x$ .
- Si  $d$  est un voisin de  $a$  (ou  $b$ ) dans les sommets de  $E$ , et qu'il est en conflit avec  $(abx)$ , alors il était nécessairement en conflit avec  $(abc)$ , car  $x$  est en conflit avec  $(abc)$  par hypothèse.

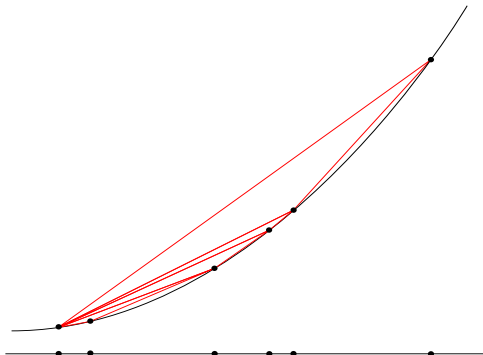
De plus un tel triangle  $(abx)$  ne peut pas non plus contenir le sommet  $d$  extérieur à  $E$  tel que  $(abd)$  soit un triangle de l'ancienne triangulation, en effet si tel était le cas, alors la triangulation du quadrilatère  $(axbd)$  serait  $(dax)$  et  $(dbx)$  et donc  $(abd)$  serait dans  $E$  ce qui par hypothèse est absurde.

## Théorème

*Construire une triangulation de Delaunay de  $n$  points peut revenir à trier un ensemble de  $n$  éléments. Il en découle que tout algorithme de triangulation de Delaunay est en  $\Omega(n \ln(n))$*

## Théorème

*Construire une triangulation de Delaunay de  $n$  points peut revenir à trier un ensemble de  $n$  éléments. Il en découle que tout algorithme de triangulation de Delaunay est en  $\Omega(n \ln n)$*



## Proposition

Le cas pire de notre algorithme incrémental est en  $O(n^3)$

## Proposition

Le cas pire de notre algorithme incrémental est en  $O(n^3)$

Dans le cas où l'ajout d'un nouveau point à gauche modifie l'intégralité de la triangulation. Le DAG est alors de hauteur  $n$ , et donc la construction complète du DAG pour l'ajout d'un points peut prendre un temps proportionnel à  $n^2$ .



TrouverTriangleContenant( $x$ ) est un parcours du DAG, sa complexité est donc relative à la hauteur du DAG, qui est en  $O(\log n)$  en moyenne et en  $O(n)$  dans le cas pire, ainsi qu'à l'arité maximale d'un sommet.

Puisqu'un triangle peut avoir tous les triangles construits pour fils, et que tous les triangles peuvent être incompatibles, cette arité est majorée par  $2p + 1$  à l'étape  $p$ , soit une complexité en  $O(n^2)$  dans le cas le pire. En moyenne, on forme  $k + 2$  nouveaux triangles à chaque étape,  $k$  étant le nombre de voisins incompatibles.

Ce nombre  $k$  sera le nombre de voisins de  $x$  dans la triangulation de Delaunay finale, il est donc en moyenne inférieur à 6 : c'est une constante. La correction de l'algorithme à un rôle prépondérant dans la complexité ! Dans RearrangerTriangulation on crée  $k + 2$  triangles et on en supprime  $k$ , chaque opération se faisant en temps constant. Le difficulté réside dans la construction du DAG, puisque le nombre d'arêtes créées à chaque étape est en  $O(k^2)$ . Dans le cas le pire, RearrangerTriangulation est donc en  $O(n^2)$ .

Puisqu'on effectue  $n$  étapes, notre algorithme a une complexité en  $O(n^3)$ , et tous les algorithmes présentés ne font pas mieux, dans la mesure où  $k$  ne peut être majoré que par  $n$ . Cependant,  $k$  est sensiblement indépendant de  $n$  puisqu'il correspond à une structure locale de la triangulation, et on peut affirmer que  $k^2$  sera lui aussi constant en moyenne. On peut donc considérer un "cas pire intermédiaire" où  $k$  est une constante et où le DAG est de hauteur  $n$ , pour une complexité en  $O(n^2)$ , qui se produit dans le cas où tous les points sont sur une parabole et où on rajoute les points dans l'ordre de descente de la parabole. Il ne serait pas rigoureux d'affirmer tel quel que des cas pires encore n'ont aucune chance de se produire, mais c'est une idée globalement partagée, probablement juste, et en tout cas très proche de la réalité. La complexité en moyenne est donc en  $O(nk \log n + k^2 n)$ , soit  $O(n \log n)$  par le même argument. En pratique, l'algorithme est effectivement assez proche de l'optimal...

Ouverture sur l'Algorithme "Divide Conquer" :

On classe l'ensemble  $P$  par ordre lexicographique sur les coordonnées.

---

**Fonction** DTrianguler( $P$ )

---

**si**  $\text{Card}(P) \leq 3$  **alors**

    On effectue la triangulation triviale d'un ensemble de moins de 3 éléments.

**sinon**

    On coupe  $P$  en  $P_1$  et  $P_2$  en son milieu

**renvoyer** Recoller (DTrianguler ( $P_1$ ), DTrianguler ( $P_2$ ))

---

Ouverture sur l'Algorithme "Divide Conquer" :

On classe l'ensemble  $P$  par ordre lexicographique sur les coordonnées.

---

**Fonction** DTriangler( $P$ )

---

**si**  $Card(P) \leq 3$  **alors**

    On effectue la triangulation triviale d'un ensemble de moins de 3 éléments.

**sinon**

    On coupe  $P$  en  $P_1$  et  $P_2$  en son milieu

**renvoyer** Recoller (DTriangler ( $P_1$ ), DTriangler ( $P_2$ ))

---

Ici c'est la fonction de recollement de deux triangulation qui va être extrêmement complexe à implémenter, et qui peut potentiellement modifier la triangulation déjà construite de manière importante.



Parisian master of research in computer science.  
2009.



Thomas H. Cormen.  
*Introduction à l'Algorithmique.*  
Dunod, 2004.



Olivier Devilliers.  
Delaunay et les classiques.  
2009.



Olivier Devilliers.  
Triangulation de delaunay, premières propriétés et algorithmes.  
2009.



Christophe Lemaire.  
*Triangulation de Delaunay et arbres multidimensionnels.*  
PhD thesis, Ecole des mines de Saint-Etienne, 1997.