

Géométrie Algorithmique

Données, Modèles, Programmes

3. Calcul Géométrique

Jean-Daniel Boissonnat

Collège de France

26 avril 2017

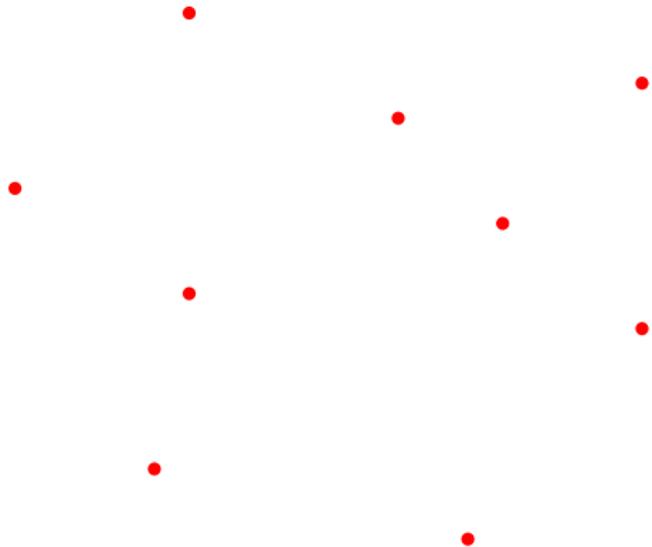
Géométrie algorithmique

Données, modèles, programmes

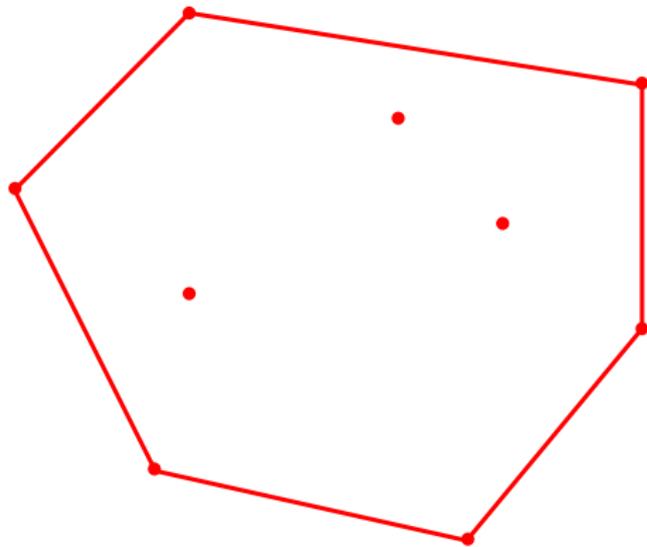
- 1 Modèles géométriques discrets
F. Cazals : Modèles géométriques pour la prédiction des interactions macro-moléculaires
- 2 La puissance de l'aléa : algorithmes randomisés
P. Calka : Probabilités géométriques
- 3 **Le calcul géométrique**
S. Pion : **La bibliothèque logicielle CGAL**
- 4 Génération de maillages
J-M. Mirebeau : Les deux réductions de Voronoï et leur application aux équations aux dérivées partielles
- 5 Courbes et surfaces
P. Alliez : Reconstruction de surfaces
- 6 Espaces de configurations
A. de Mesmay : Dessin de graphes
- 7 Structures de données géométriques
D. Feldman : Core sets
- 8 Géométrie des données
F. Chazal : Analyse topologique des données

Outline

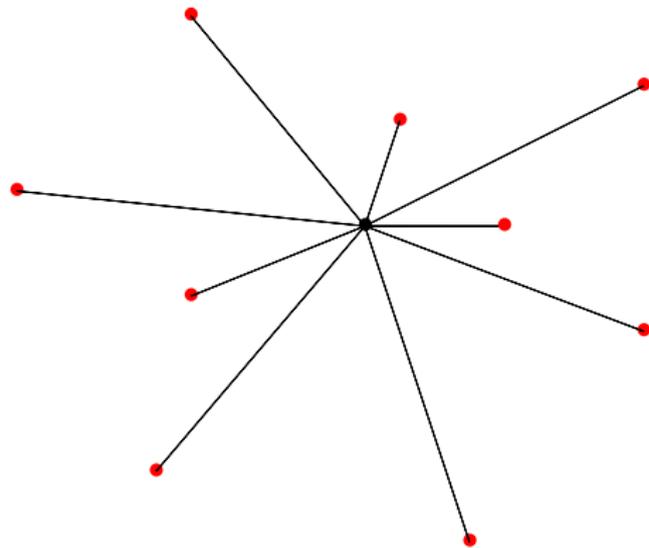
La double nature des objets géométriques



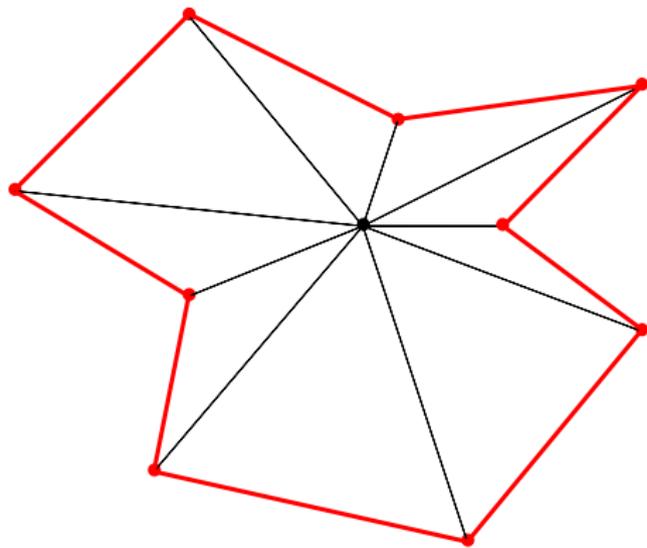
La double nature des objets géométriques



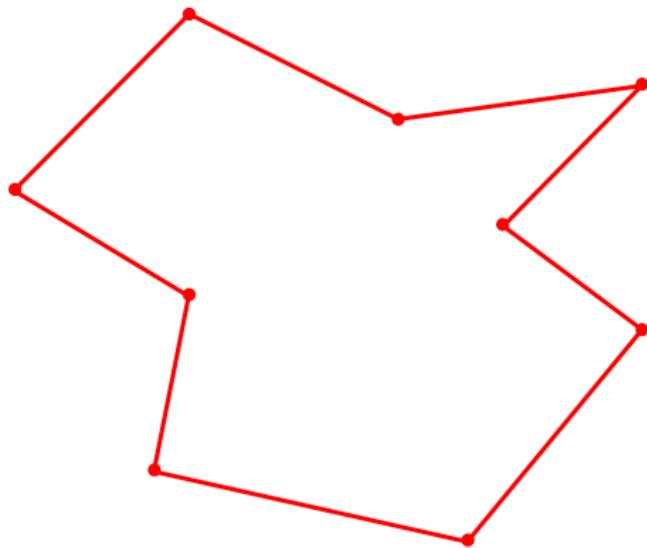
La double nature des objets géométriques



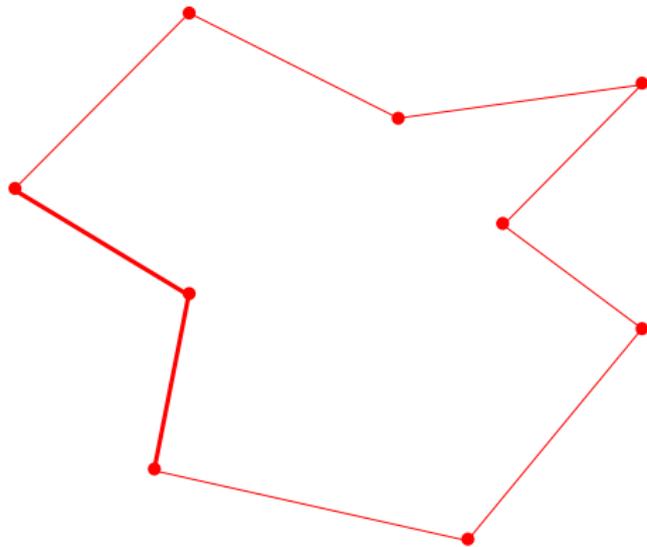
La double nature des objets géométriques



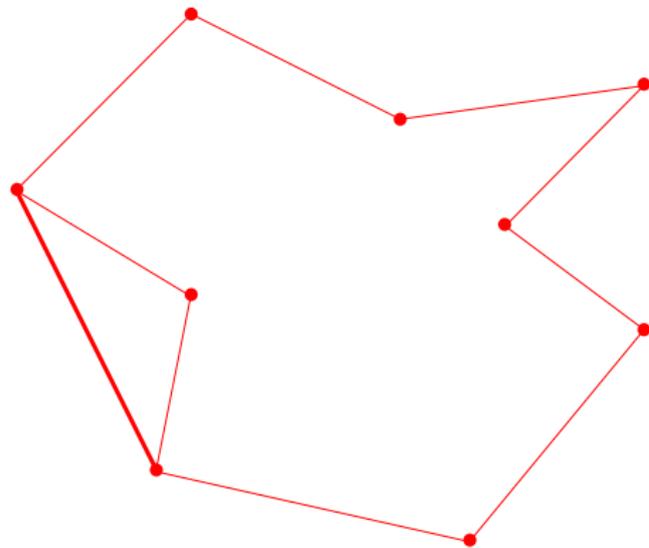
La double nature des objets géométriques



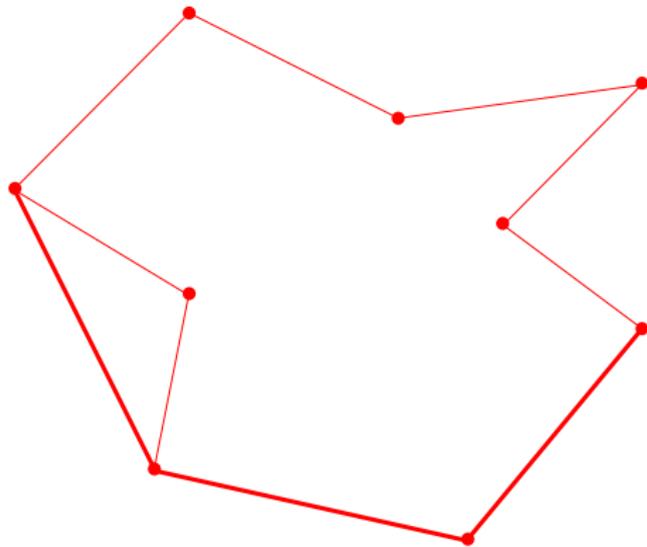
La double nature des objets géométriques



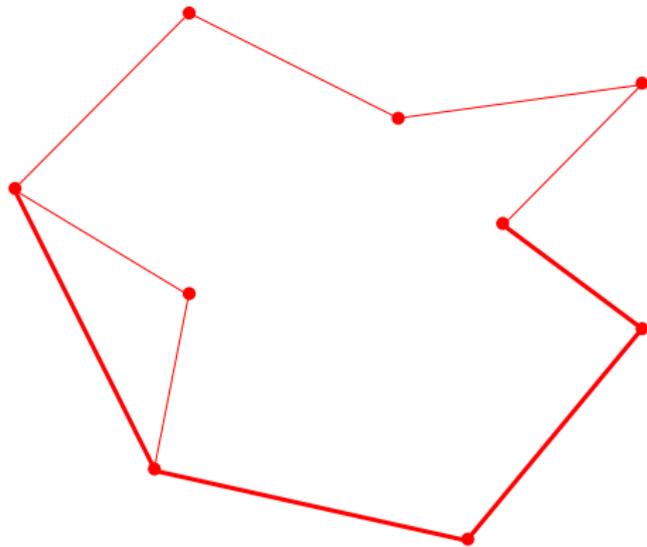
La double nature des objets géométriques



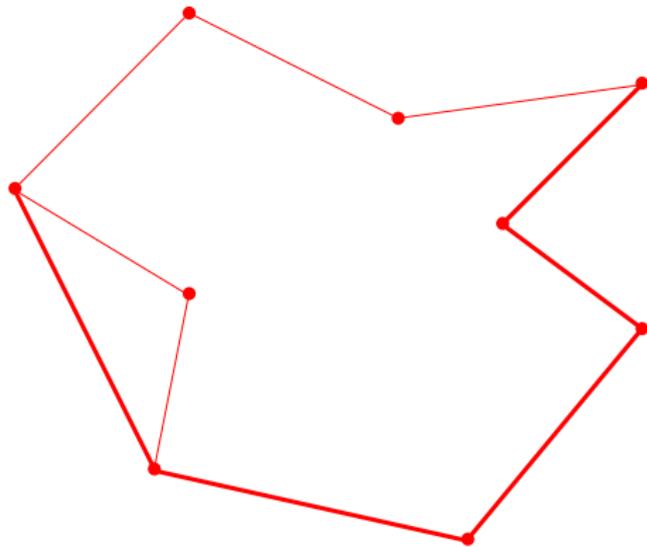
La double nature des objets géométriques



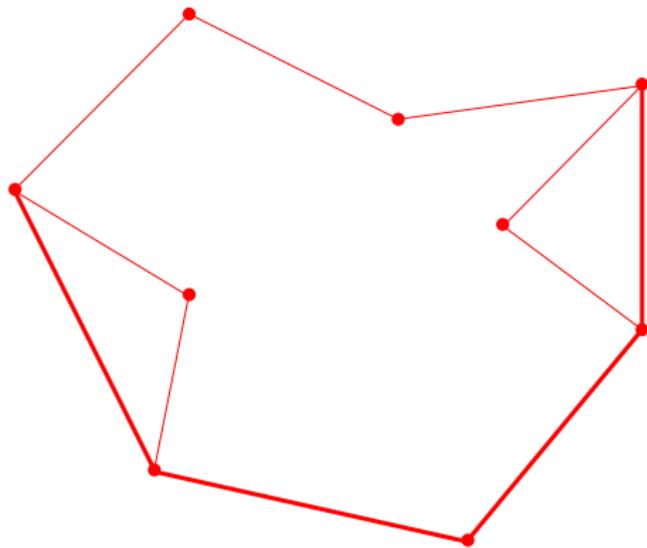
La double nature des objets géométriques



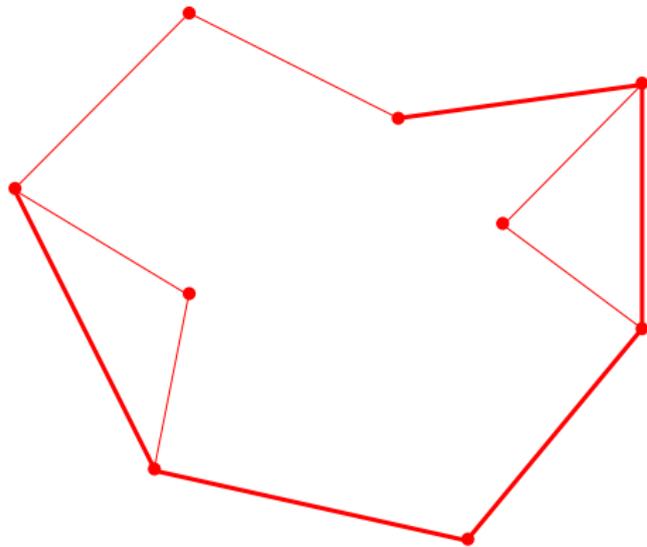
La double nature des objets géométriques



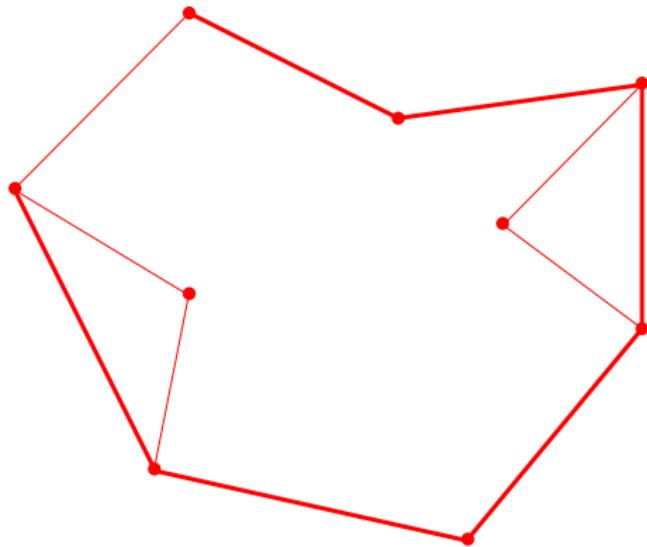
La double nature des objets géométriques



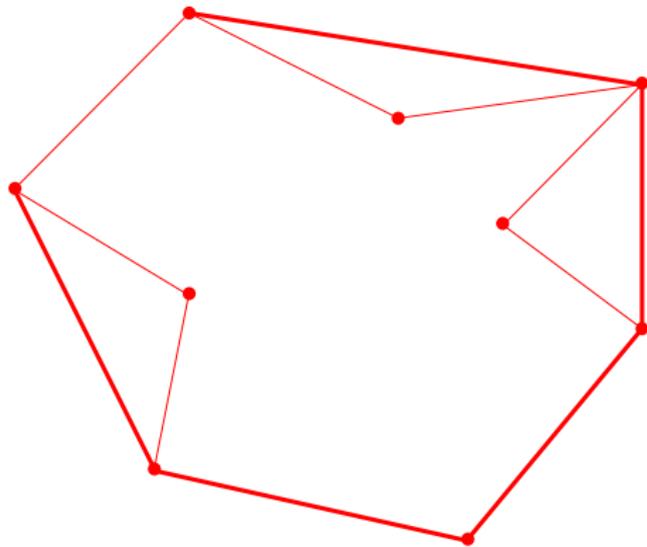
La double nature des objets géométriques



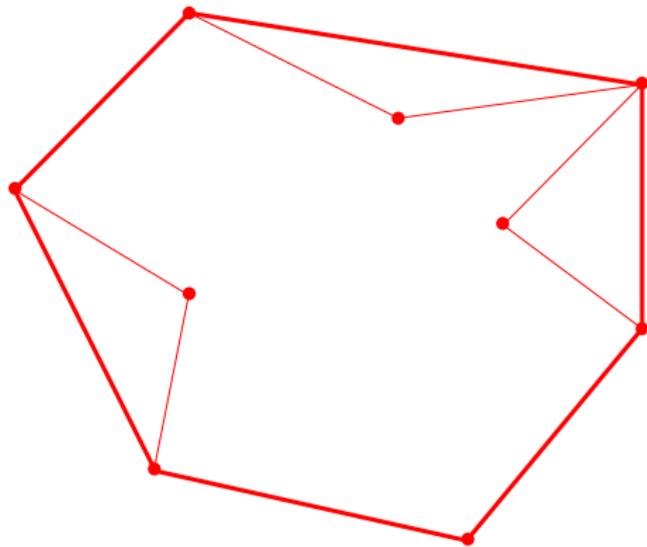
La double nature des objets géométriques



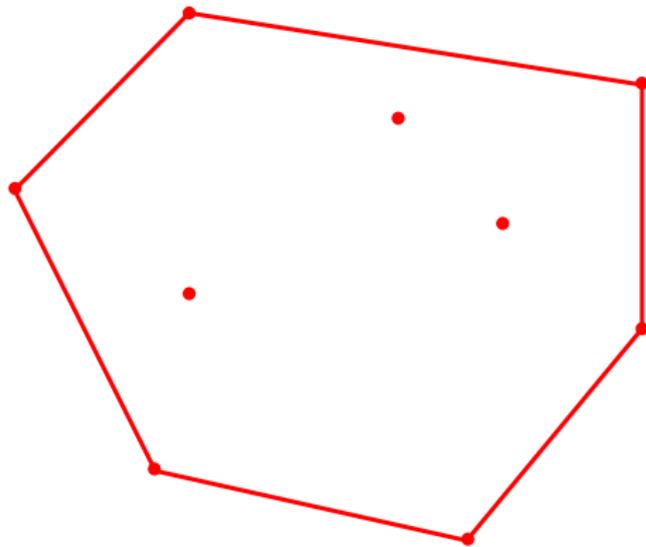
La double nature des objets géométriques



La double nature des objets géométriques



La double nature des objets géométriques



Modèles de calcul

Le modèle théorique standard

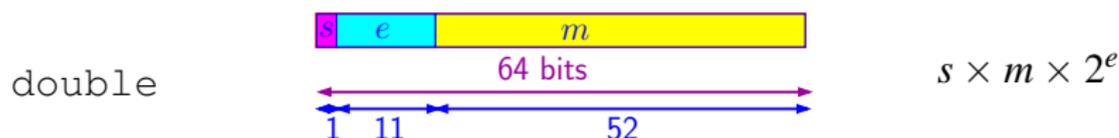
Real RAM : toutes les opérations élémentaires sur les nombres réels peuvent être exécutées exactement et en temps $O(1)$

- + Couvre tous les problèmes géométriques
- + Autorise l'utilisation des mathématiques classiques
- + A conduit à un essor spectaculaire des aspects théoriques de la GA

- Mais...

Modèles de calcul

Arithmétique des ordinateurs



- Les nombres machine des ordinateurs ont une représentation **finie** : nombres entiers ou flottants
par ex. $\sqrt{2}$ n'est pas représentable sous forme d'un nombre flottant
- Les algorithmes géométriques s'appuient sur des théorèmes géométriques qui ne sont plus vrais dans le monde discret de l'arithmétique flottante
- Les représentations de nombres algébriques (logiciel) sont comparativement très lentes

Modèles de calcul

Real RAM \neq RAM + arithmétique flottante

Mathematically, the problem of reporting intersecting segments is trivial. Computationally, the problem is far from easy, and be impossible to solve reliably and consistently.

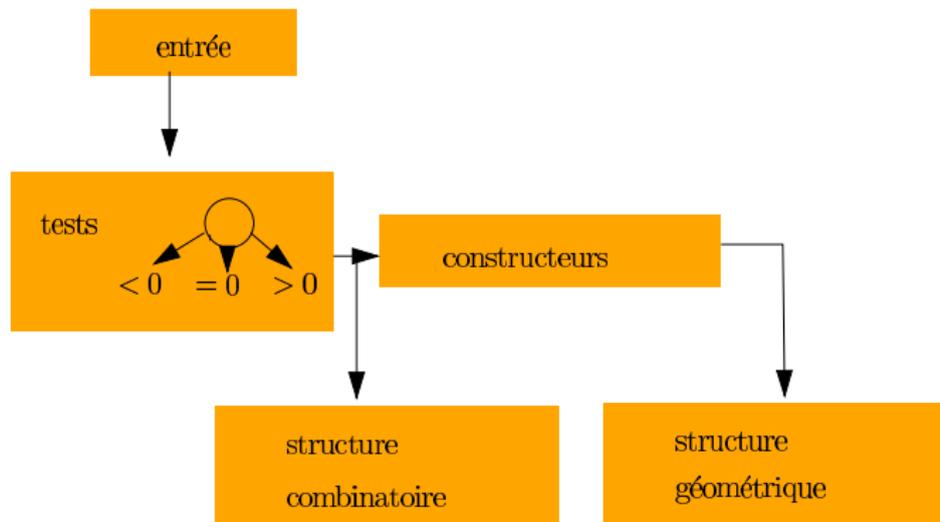
R. Forrest 1986

Dans l'industrie, la réalisation et la maintenance d'algorithmes géométriques comportant une part combinatoire et une part numérique sont très coûteuses. La réalisation de tels algorithmes ne procède pas d'une méthodologie rigoureuse et reproductible mais d'un savoir-faire personnel difficilement transmissible. Si les conséquences semblent mineures et n'affecter que des cas limites peu fréquents, en fait, on remarque que la gestion des ϵ et des cas limites devient la partie la plus coûteuse en mise au point et en maintenance.

A. Lieutier, Matra Datavision, 1997

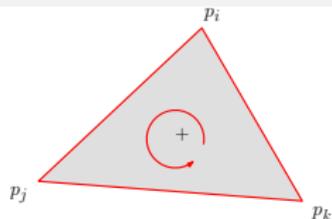
Arbre de décision

Prédicats et constructeurs



Dégénérescences et instabilités numériques

Prédicat d'orientation



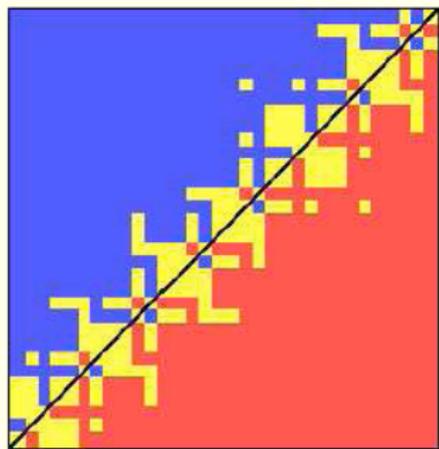
$$\text{orient}(p_0 p_1 p_2) = \text{signe} \begin{vmatrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ 1 & 1 & 1 \end{vmatrix} = \text{signe} \begin{vmatrix} x_1 - x_0 & x_2 - x_0 \\ y_1 - y_0 & y_2 - y_0 \end{vmatrix}$$

$$\begin{aligned} p &= (0.5 + x.u, 0.5 + y.u) \\ 0 \leq x, y &< 256, \quad u = 2^{-53} \\ q &= (12, 12) \\ r &= (24, 24) \end{aligned}$$

$\text{orientation}(p, q, r)$
evaluated with double

256 x 256 pixel image

> 0, = 0, < 0

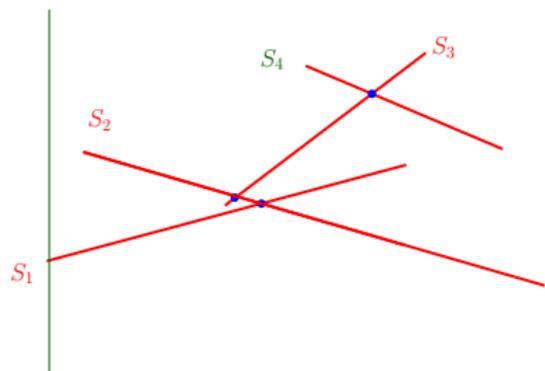


Kettner et al. [2008]

Arrangement de segments du plan

Algorithme de balayage [Bentley-Ottman 1979]

Balaie le plan par une droite verticale Δ allant de gauche à droite



- la liste des segments coupés par Δ ne change qu'en certains évènements (extrémités et intersections)
- les intersections sont découvertes à la volée : si 2 segments se coupent en I , ils sont consécutifs dans la liste à gauche de I

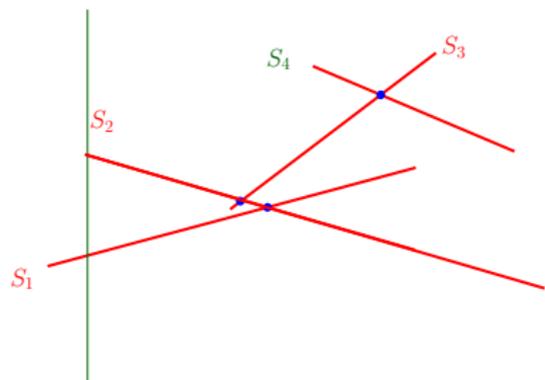
On maintient

- une queue de priorité X d'évènements triés en x (initialisée avec les extrémités des segments)
- un dictionnaire Y des segments coupés par Δ triés selon l'ordonnée de leur intersection avec Δ

Arrangement de segments du plan

Algorithme de balayage [Bentley-Ottman 1979]

Balaie le plan par une droite verticale Δ allant de gauche à droite



- la liste des segments coupés par Δ ne change qu'en certains évènements (extrémités et intersections)
- les intersections sont découvertes à la volée : si 2 segments se coupent en I , ils sont consécutifs dans la liste à gauche de I

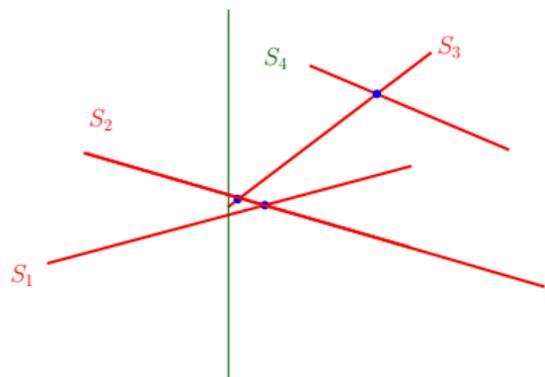
On maintient

- une queue de priorité X d'évènements triés en x (initialisée avec les extrémités des segments)
- un dictionnaire Y des segments coupés par Δ triés selon l'ordonnée de leur intersection avec Δ

Arrangement de segments du plan

Algorithme de balayage [Bentley-Ottman 1979]

Balaie le plan par une droite verticale Δ allant de gauche à droite



- la liste des segments coupés par Δ ne change qu'en certains évènements (extrémités et intersections)
- les intersections sont découvertes à la volée : si 2 segments se coupent en I , ils sont consécutifs dans la liste à gauche de I

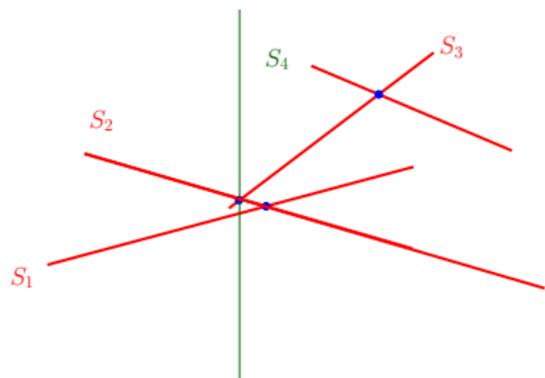
On maintient

- une queue de priorité X d'évènements triés en x (initialisée avec les extrémités des segments)
- un dictionnaire Y des segments coupés par Δ triés selon l'ordonnée de leur intersection avec Δ

Arrangement de segments du plan

Algorithme de balayage [Bentley-Ottman 1979]

Balaie le plan par une droite verticale Δ allant de gauche à droite



- la liste des segments coupés par Δ ne change qu'en certains évènements (extrémités et intersections)
- les intersections sont découvertes à la volée : si 2 segments se coupent en I , ils sont consécutifs dans la liste à gauche de I

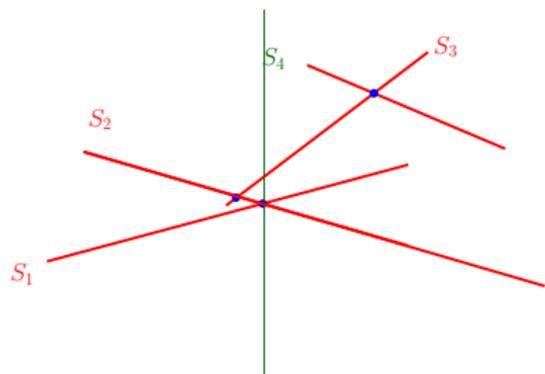
On maintient

- une queue de priorité X d'évènements triés en x (initialisée avec les extrémités des segments)
- un dictionnaire Y des segments coupés par Δ triés selon l'ordonnée de leur intersection avec Δ

Arrangement de segments du plan

Algorithme de balayage [Bentley-Ottman 1979]

Balaie le plan par une droite verticale Δ allant de gauche à droite



- la liste des segments coupés par Δ ne change qu'en certains évènements (extrémités et intersections)
- les intersections sont découvertes à la volée : si 2 segments se coupent en I , ils sont consécutifs dans la liste à gauche de I

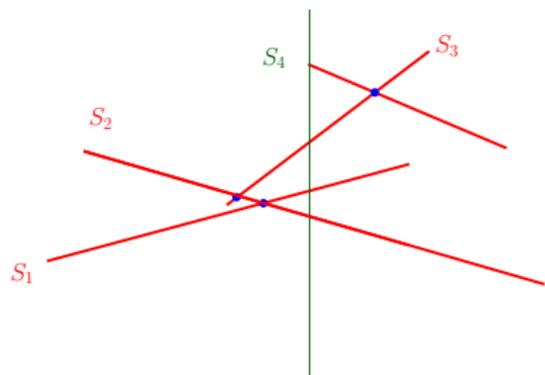
On maintient

- une queue de priorité X d'évènements triés en x (initialisée avec les extrémités des segments)
- un dictionnaire Y des segments coupés par Δ triés selon l'ordonnée de leur intersection avec Δ

Arrangement de segments du plan

Algorithme de balayage [Bentley-Ottman 1979]

Balaie le plan par une droite verticale Δ allant de gauche à droite



- la liste des segments coupés par Δ ne change qu'en certains évènements (extrémités et intersections)
- les intersections sont découvertes à la volée : si 2 segments se coupent en I , ils sont consécutifs dans la liste à gauche de I

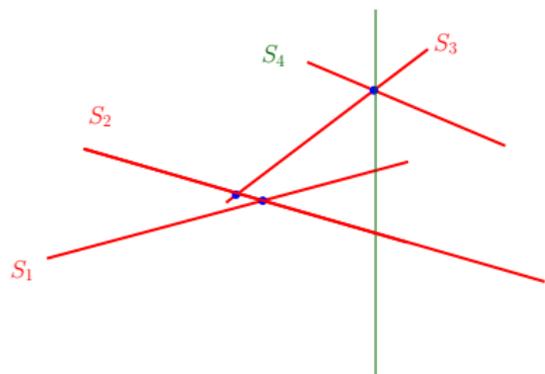
On maintient

- une queue de priorité X d'évènements triés en x (initialisée avec les extrémités des segments)
- un dictionnaire Y des segments coupés par Δ triés selon l'ordonnée de leur intersection avec Δ

Arrangement de segments du plan

Algorithme de balayage [Bentley-Ottman 1979]

Balaie le plan par une droite verticale Δ allant de gauche à droite



- la liste des segments coupés par Δ ne change qu'en certains évènements (extrémités et intersections)
- les intersections sont découvertes à la volée : si 2 segments se coupent en I , ils sont consécutifs dans la liste à gauche de I

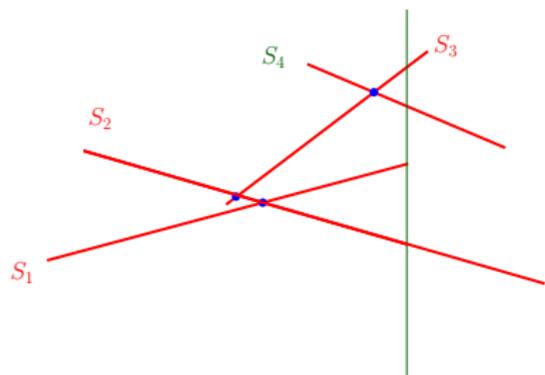
On maintient

- une queue de priorité X d'évènements triés en x (initialisée avec les extrémités des segments)
- un dictionnaire Y des segments coupés par Δ triés selon l'ordonnée de leur intersection avec Δ

Arrangement de segments du plan

Algorithme de balayage [Bentley-Ottman 1979]

Balaie le plan par une droite verticale Δ allant de gauche à droite



- la liste des segments coupés par Δ ne change qu'en certains évènements (extrémités et intersections)
- les intersections sont découvertes à la volée : si 2 segments se coupent en I , ils sont consécutifs dans la liste à gauche de I

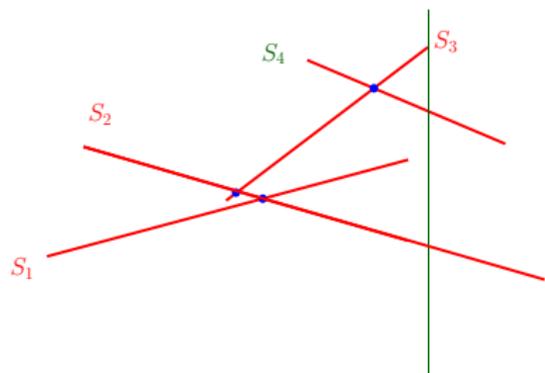
On maintient

- une queue de priorité X d'évènements triés en x (initialisée avec les extrémités des segments)
- un dictionnaire Y des segments coupés par Δ triés selon l'ordonnée de leur intersection avec Δ

Arrangement de segments du plan

Algorithme de balayage [Bentley-Ottman 1979]

Balaie le plan par une droite verticale Δ allant de gauche à droite



- la liste des segments coupés par Δ ne change qu'en certains évènements (extrémités et intersections)
- les intersections sont découvertes à la volée : si 2 segments se coupent en I , ils sont consécutifs dans la liste à gauche de I

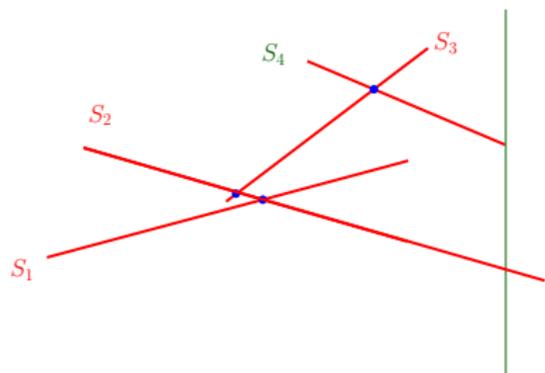
On maintient

- une queue de priorité X d'évènements triés en x (initialisée avec les extrémités des segments)
- un dictionnaire Y des segments coupés par Δ triés selon l'ordonnée de leur intersection avec Δ

Arrangement de segments du plan

Algorithme de balayage [Bentley-Ottman 1979]

Balaie le plan par une droite verticale Δ allant de gauche à droite



- la liste des segments coupés par Δ ne change qu'en certains évènements (extrémités et intersections)
- les intersections sont découvertes à la volée : si 2 segments se coupent en I , ils sont consécutifs dans la liste à gauche de I

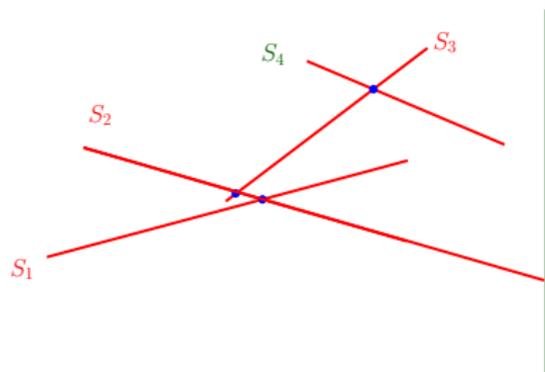
On maintient

- une queue de priorité X d'évènements triés en x (initialisée avec les extrémités des segments)
- un dictionnaire Y des segments coupés par Δ triés selon l'ordonnée de leur intersection avec Δ

Arrangement de segments du plan

Algorithme de balayage [Bentley-Ottman 1979]

Balaie le plan par une droite verticale Δ allant de gauche à droite



- la liste des segments coupés par Δ ne change qu'en certains évènements (extrémités et intersections)
- les intersections sont découvertes à la volée : si 2 segments se coupent en I , ils sont consécutifs dans la liste à gauche de I

On maintient

- une queue de priorité X d'évènements triés en x (initialisée avec les extrémités des segments)
- un dictionnaire Y des segments coupés par Δ triés selon l'ordonnée de leur intersection avec Δ

Arrangement de segments du plan

Analyse de l'algorithme de balayage

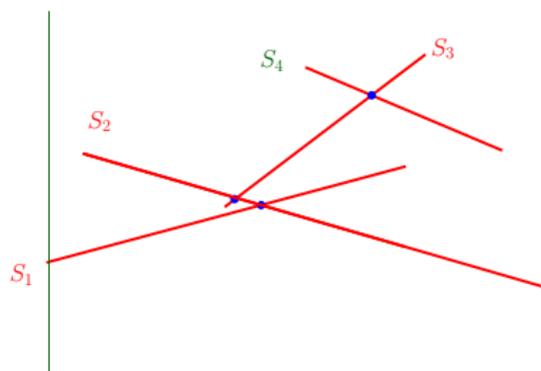
Complexité

- Tri initial : $O(n \log n)$
- L'algorithme s'arrête sur $2n + k$ évènements
 - en chacun d'eux, il effectue $O(1)$ accès à X et Y
 - qui chacun coûte $O(\log n)$ (arbres équilibrés)

On peut calculer un arrangement de n segments de droite dont k paires se coupent en temps $O((n + k) \log n)$

Se généralise au cas d'arcs de courbes monotones (sans tangente verticale)

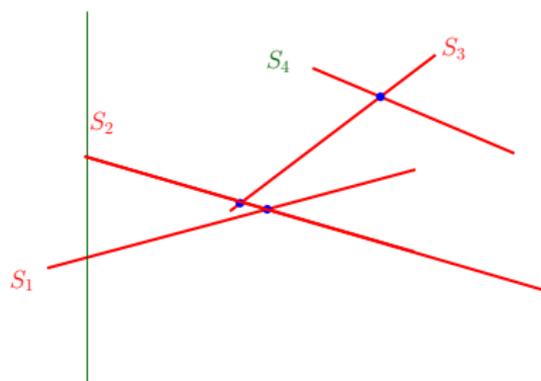
Une erreur numérique fatale



| | | | | | | | | | | |
|-------|-------|-------|------------------|-------|----------|-------|-------|-------|-------|-------|
| $Y :$ | S_1 | S_2 | S_1 | S_1 | S_1 | S_4 | S_4 | S_4 | S_2 | |
| | | S_1 | S_2 | S_2 | S_3 | S_1 | S_3 | S_2 | | |
| | | | S_3 | S_3 | S_2 | S_3 | S_2 | | | |
| | | | | | | S_2 | | | | |
| $X :$ | A_1 | A_2 | \tilde{I}_{12} | A_3 | I_{23} | A_4 | B_1 | B_3 | B_4 | B_2 |

$$S_3 \cap S_4 = \emptyset !!$$

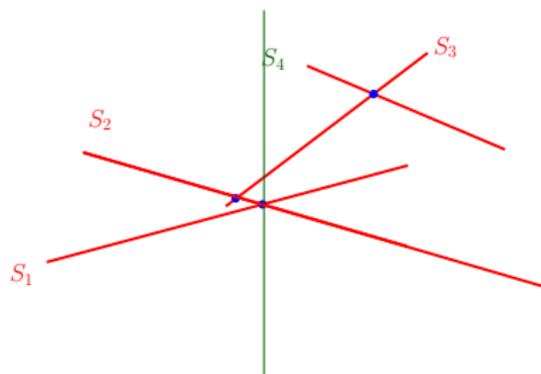
Une erreur numérique fatale



| | | | | | | | | | | |
|-------|-------|----------------|------------------|-------------------------|-------------------------|----------------------------------|-------------------------|----------------|-------|-------|
| $Y :$ | S_1 | S_2 S_1 | S_1 S_2 | S_1 S_2 S_3 | S_1 S_3 S_2 | S_4 S_1 S_3 S_2 | S_4 S_3 S_2 | S_4 S_2 | S_2 | |
| $X :$ | A_1 | A_2 | \tilde{I}_{12} | A_3 | I_{23} | A_4 | B_1 | B_3 | B_4 | B_2 |

$$S_3 \cap S_4 = \emptyset !!$$

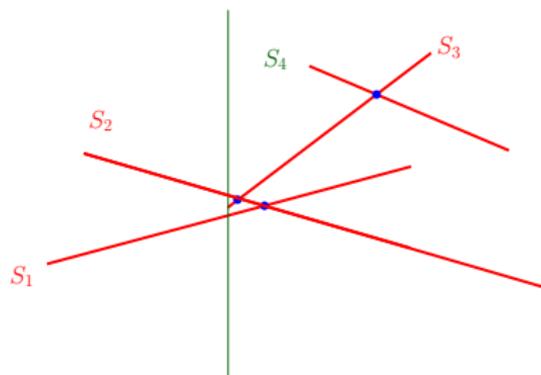
Une erreur numérique fatale



| | | | | | | | | | | |
|-------|-------|-------|------------------|-------|----------|-------|-------|-------|-------|-------|
| $Y :$ | S_1 | S_2 | S_1 | S_1 | S_1 | S_4 | S_4 | S_4 | S_2 | |
| | | S_1 | S_2 | S_2 | S_3 | S_1 | S_3 | S_2 | | |
| | | | S_3 | S_3 | S_2 | S_3 | S_2 | | | |
| | | | | | | S_2 | | | | |
| $X :$ | A_1 | A_2 | \tilde{I}_{12} | A_3 | I_{23} | A_4 | B_1 | B_3 | B_4 | B_2 |

$$S_3 \cap S_4 = \emptyset !!$$

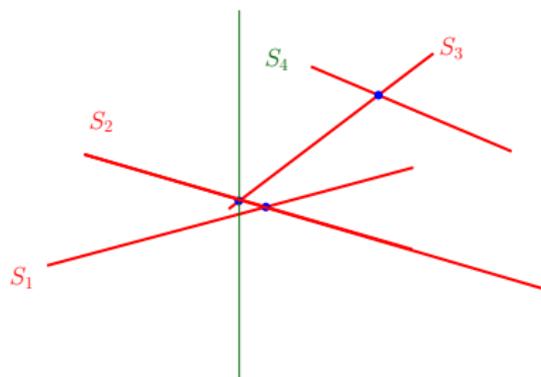
Une erreur numérique fatale



| | | | | | | | | | | |
|-------|-------|-------|------------------|-------|----------|-------|-------|-------|-------|-------|
| $Y :$ | S_1 | S_2 | S_1 | S_1 | S_1 | S_4 | S_4 | S_4 | S_2 | |
| | | S_1 | S_2 | S_2 | S_3 | S_1 | S_3 | S_2 | | |
| | | | S_3 | S_3 | S_2 | S_3 | S_2 | | | |
| $X :$ | A_1 | A_2 | \tilde{I}_{12} | A_3 | I_{23} | A_4 | B_1 | B_3 | B_4 | B_2 |

$$S_3 \cap S_4 = \emptyset !!$$

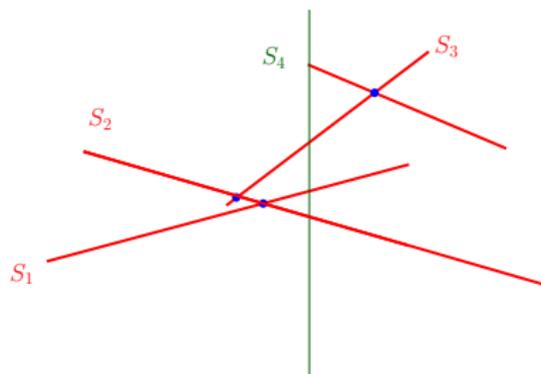
Une erreur numérique fatale



| | | | | | | | | | | |
|-------|-------|----------------|------------------|-------------------------|-------------------------|----------------------------------|-------------------------|----------------|-------|-------|
| $Y :$ | S_1 | S_2 S_1 | S_1 S_2 | S_1 S_2 S_3 | S_1 S_3 S_2 | S_4 S_1 S_3 S_2 | S_4 S_3 S_2 | S_4 S_2 | S_2 | |
| $X :$ | A_1 | A_2 | \tilde{I}_{12} | A_3 | I_{23} | A_4 | B_1 | B_3 | B_4 | B_2 |

$$S_3 \cap S_4 = \emptyset !!$$

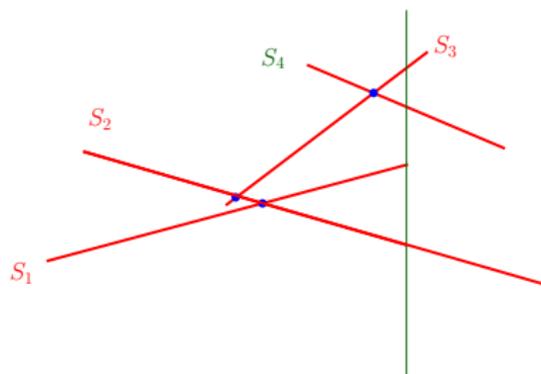
Une erreur numérique fatale



| | | | | | | | | | | |
|-------|-------|-------|------------------|-------|----------|-------|-------|-------|-------|-------|
| $Y :$ | S_1 | S_2 | S_1 | S_1 | S_1 | S_4 | S_4 | S_4 | S_2 | |
| | | S_1 | S_2 | S_2 | S_3 | S_1 | S_3 | S_2 | | |
| | | | S_3 | S_3 | S_2 | S_3 | S_2 | | | |
| | | | | | | S_2 | | | | |
| $X :$ | A_1 | A_2 | \tilde{I}_{12} | A_3 | I_{23} | A_4 | B_1 | B_3 | B_4 | B_2 |

$$S_3 \cap S_4 = \emptyset !!$$

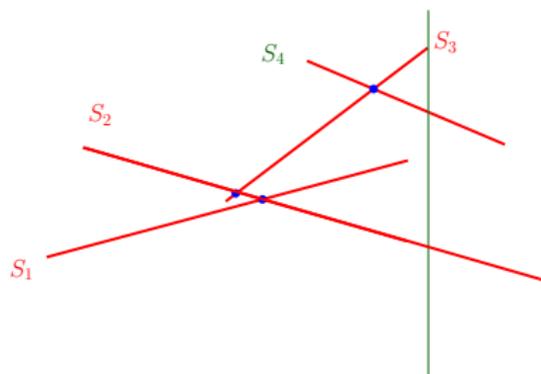
Une erreur numérique fatale



| | | | | | | | | | | |
|-------|-------|-------|------------------|-------|----------|-------|-------|-------|-------|-------|
| $Y :$ | S_1 | S_2 | S_1 | S_1 | S_1 | S_4 | S_4 | S_4 | S_2 | |
| | | S_1 | S_2 | S_2 | S_3 | S_1 | S_3 | S_2 | | |
| | | | S_3 | S_3 | S_2 | S_3 | S_2 | | | |
| | | | | | | S_2 | | | | |
| $X :$ | A_1 | A_2 | \tilde{I}_{12} | A_3 | I_{23} | A_4 | B_1 | B_3 | B_4 | B_2 |

$$S_3 \cap S_4 = \emptyset !!$$

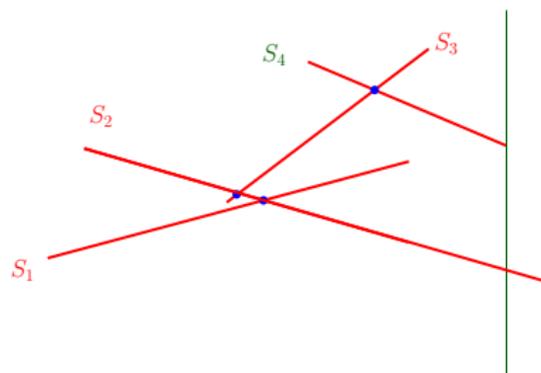
Une erreur numérique fatale



| | | | | | | | | | | |
|-------|-------|-------|------------------|-------|----------|-------|-------|-------|-------|-------|
| $Y :$ | S_1 | S_2 | S_1 | S_1 | S_1 | S_4 | S_4 | S_4 | S_2 | |
| | | S_1 | S_2 | S_2 | S_3 | S_1 | S_3 | S_2 | | |
| | | | S_3 | S_3 | S_2 | S_3 | S_2 | | | |
| | | | | | | S_2 | | | | |
| $X :$ | A_1 | A_2 | \tilde{I}_{12} | A_3 | I_{23} | A_4 | B_1 | B_3 | B_4 | B_2 |

$$S_3 \cap S_4 = \emptyset !!$$

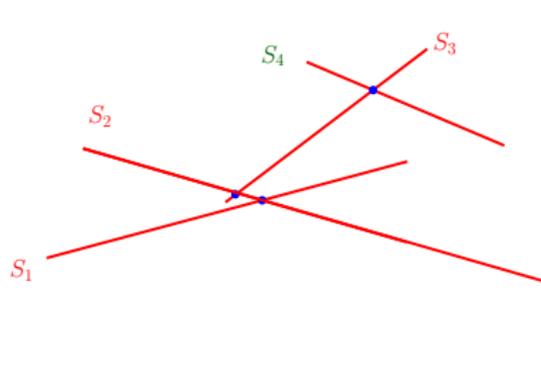
Une erreur numérique fatale



| | | | | | | | | | | |
|-------|-------|-------|------------------|-------|----------|-------|-------|-------|-------|-------|
| $Y :$ | S_1 | S_2 | S_1 | S_1 | S_1 | S_4 | S_4 | S_4 | S_2 | |
| | | S_1 | S_2 | S_2 | S_3 | S_1 | S_3 | S_2 | | |
| | | | S_3 | S_3 | S_2 | S_3 | S_2 | | | |
| | | | | | | S_2 | | | | |
| $X :$ | A_1 | A_2 | \tilde{I}_{12} | A_3 | I_{23} | A_4 | B_1 | B_3 | B_4 | B_2 |

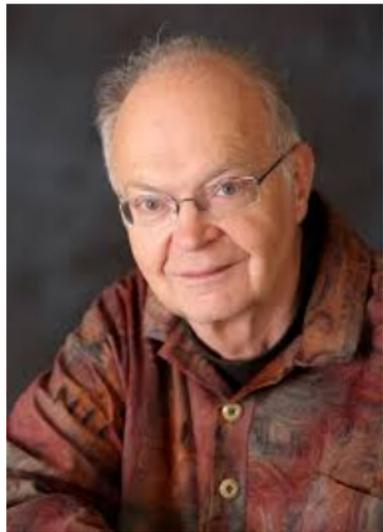
$$S_3 \cap S_4 = \emptyset !!$$

Une erreur numérique fatale



| | | | | | | | | | | |
|-------|-------|-------|------------------|-------|----------|-------|-------|-------|-------|-------|
| $Y :$ | S_1 | S_2 | S_1 | S_1 | S_1 | S_4 | S_4 | S_4 | S_2 | |
| | | S_1 | S_2 | S_2 | S_3 | S_1 | S_3 | S_2 | | |
| | | | S_3 | S_3 | S_2 | S_3 | S_2 | | | |
| | | | | | | S_2 | | | | |
| $X :$ | A_1 | A_2 | \tilde{I}_{12} | A_3 | I_{23} | A_4 | B_1 | B_3 | B_4 | B_2 |

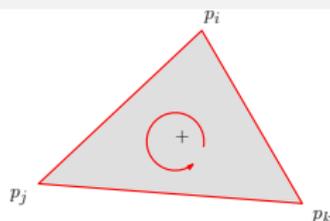
$$S_3 \cap S_4 = \emptyset !!$$



Although the methods reside perhaps on a low level, they underlie all of the more grandiose applications of computers to numerical problems. We are concerned here with the interface between numerical mathematics and computer programming, and it is the mating of both types of skills that makes the subject so interesting.

D. Knuth, 1980

Le prédicat d'orientation



$$\text{orient}(p_0 p_1 p_2) = \text{signe} \begin{vmatrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ 1 & 1 & 1 \end{vmatrix} = \text{signe} \begin{vmatrix} x_1 - x_0 & x_2 - x_0 \\ y_1 - y_0 & y_2 - y_0 \end{vmatrix}$$

en dimensions supérieures,

$$\text{orient}(p_0 \dots p_d) = \text{sign} \begin{vmatrix} p_0 & \dots & p_d \\ 1 & \dots & 1 \end{vmatrix} = \text{sign} |p_1 - p_0 \dots p_d - p_0|$$

Il faut évaluer le signe d'un polynôme irréductible de **degré d** en les variables d'entrée

Degré algébrique des prédicats

[Burnikel 96], [Liotta et al. 97]

input = variable

prédicat = signe $-$, 0 , $+$ d'un polynôme multivariable

fonction booléenne de prédicats élémentaires

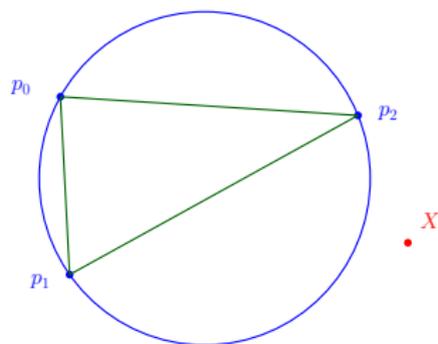
degré d'un algorithme A

= **max** degrés des facteurs irréduc. de signes non cst
des polynômes associés aux prédicats évalués by A

degré d'un problème P

= **min** degrés de tout algorithme résolvant P

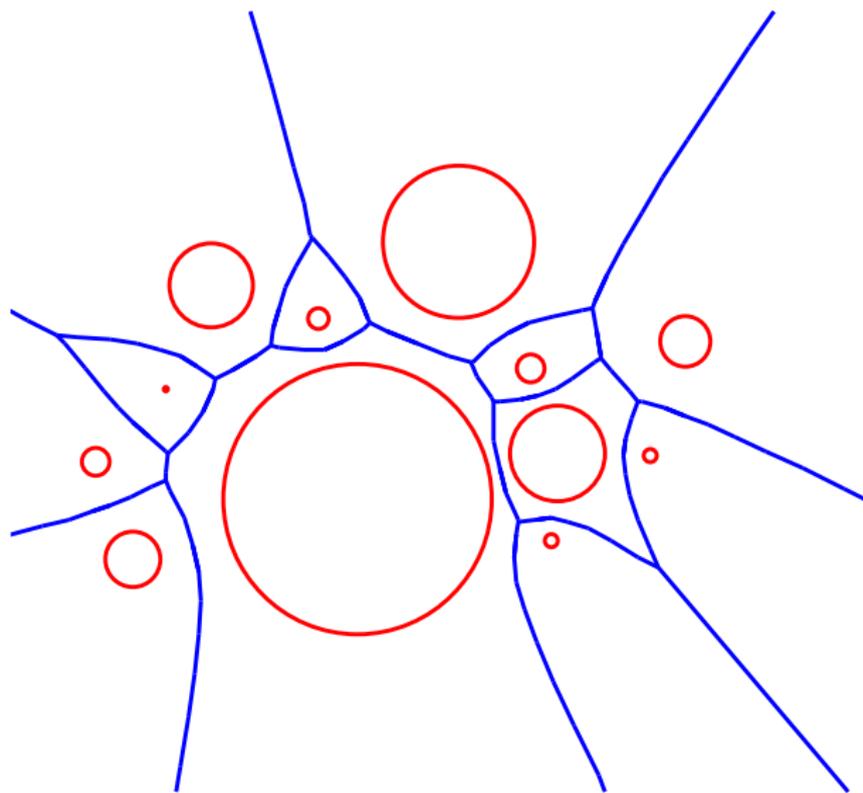
Prédicat InCircle



$$\text{InCircle}(p_0, p_1, p_2, X) = \text{signe} \begin{vmatrix} x_0 & x_1 & x_2 & x \\ y_0 & y_1 & y_2 & y \\ x_0^2 + y_0^2 & x_1^2 + y_1^2 & x_2^2 + y_2^2 & x^2 + y^2 \\ 1 & 1 & 1 & 1 \end{vmatrix}$$

degré : $d + 2$

Diagramme de Voronoï de sphères



Degré : 16

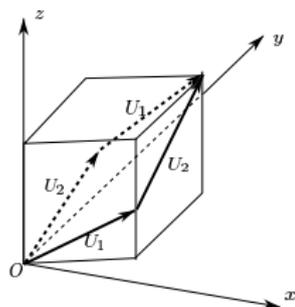
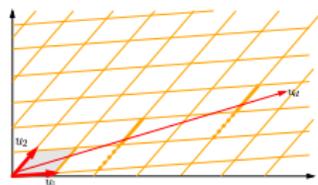
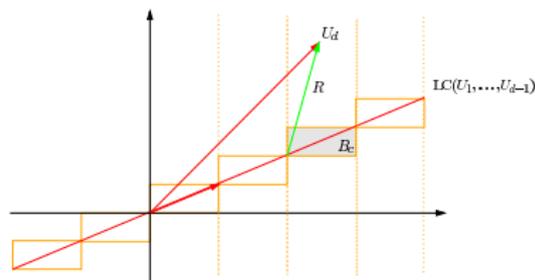
Evaluation exacte de prédicats

- Garantit l'exactitude de la structure combinatoire
- De nombreuses approches fournissent des solutions efficaces aux problèmes de petit degré
 - ▶ Évaluation de signes de déterminants en simple précision
 - ▶ Bornes de séparation de racines de polynômes
 - ▶ Calcul multi-précision (arithmétique modulaire, expansions flottantes, filtres arithmétiques)
- Elle est implantée dans **CGAL** !

Evaluation en précision fixe de signes de déterminants

[Boissonnat et al. 92] [Brönnimann & Yvinec 97]

- $D = \det(U_1, \dots, U_d)$
- Les $U_i, i \leq d - 1$, définissent un réseau dans l'hyp. (O, U_1, \dots, U_{d-1})
- A chaque cellule de ce réseau, on associe une boîte $B_c = c \oplus U_1 \oplus \dots \oplus U_{d-1} \oplus \sum_{i=1}^{d-1} z_i E_d$

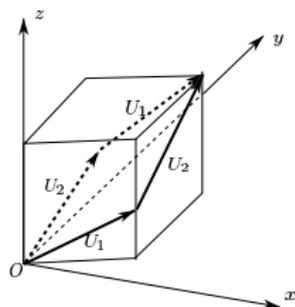
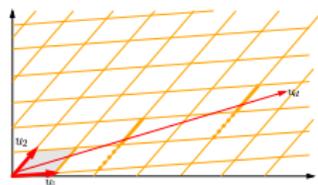
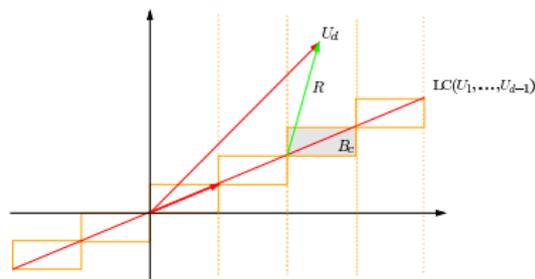


$$D = \det(U_1, \dots, U_{d-1}, R) \quad \text{avec} \quad R = U_d - c$$

Evaluation en précision fixe de signes de déterminants

[Boissonnat et al. 92] [Brönnimann & Yvinec 97]

- $D = \det(U_1, \dots, U_d)$
- Les $U_i, i \leq d - 1$, définissent un réseau dans l'hyp. (O, U_1, \dots, U_{d-1})
- A chaque cellule de ce réseau, on associe une boîte $B_c = c \oplus U_1 \oplus \dots \oplus U_{d-1} \oplus \sum_{i=1}^{d-1} z_i E_d$

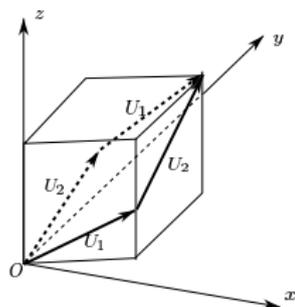
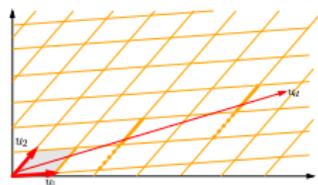
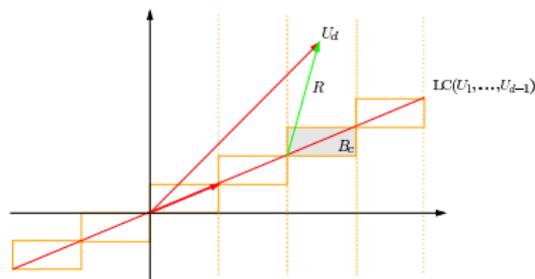


$$D = \det(U_1, \dots, U_{d-1}, R) \quad \text{avec} \quad R = U_d - c$$

Evaluation en précision fixe de signes de déterminants

[Boissonnat et al. 92] [Brönnimann & Yvinec 97]

- $D = \det(U_1, \dots, U_d)$
- Les $U_i, i \leq d - 1$, définissent un réseau dans l'hyp. (O, U_1, \dots, U_{d-1})
- A chaque cellule de ce réseau, on associe une boîte $B_c = c \oplus U_1 \oplus \dots \oplus U_{d-1} \oplus \sum_{i=1}^{d-1} z_i E_d$

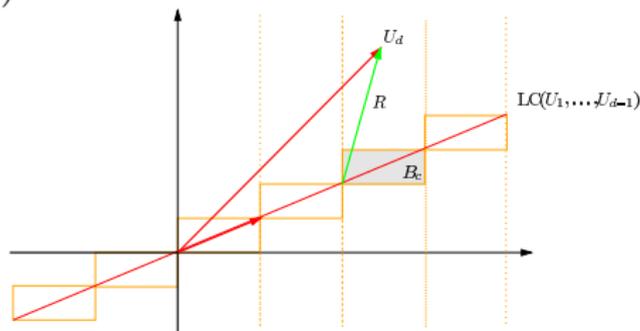


$$D = \det(U_1, \dots, U_{d-1}, R) \quad \text{avec} \quad R = U_d - c$$

Evaluation en précision fixe de signes de déterminants

[Boissonnat et al. 92] [Brönnimann & Yvinec 97]

$$D = \det(U_1, \dots, U_d)$$



1. **localiser** (U_d, c) : Identifier la cellule $C \ni u_d = \text{proj}(U_d)$
2. $R := U_d - \text{origine}(B_c)$;
3. **si** $z(U_d) > z^+(B_c)$ **alors** $\text{sign}(D) := \text{sign}(u_1, \dots, u_{d-1})$
si $z(U_d) < z^-(B_c)$ **alors** $\text{sign}(D) := -\text{sign}(u_1, \dots, u_{d-1})$
sinon
 $R := 2R$
 localiser (R, c)
 goto 3

Analyse de l'algorithme 1/2

Nombre maximal d'itérations

$$\begin{aligned} \text{Chaque composante d'un } U_i \text{ est } \leq 2^b &\Rightarrow \|U_i\| \leq \sqrt{d} 2^b \\ R_{it} \leq d 2^b &\Rightarrow \|R_{it}\| \leq d \sqrt{d} 2^b \end{aligned}$$

$$\begin{aligned} |D_{it}| = 2^{it} D &= |U_1, U_2, \dots, R_{it}| \\ &\leq \|U_1\| \times \dots \times \|U_{d-1}\| \times \|R_{it}\| \quad (\text{Hadamard}) \\ &\leq d \sqrt{d}^d 2^{bd} \end{aligned}$$

$$D \neq 0 \Rightarrow |D| \geq 1 \Rightarrow \text{it} \leq bd + \left\lceil \left(\frac{d}{2} + 1\right) \log d \right\rceil$$

Analyse de l'algorithme 2/2

Complexité

cas le pire :

$$T = O(b^d (d!)^2)$$

bien mieux en pratique, même si le déterminant est nul

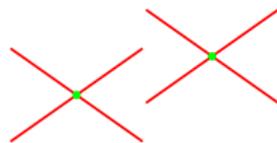
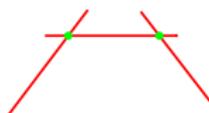
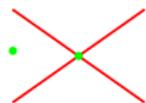
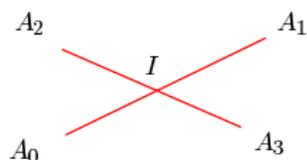
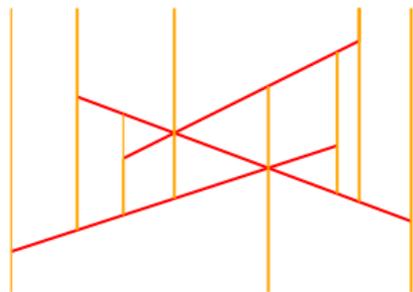
entrées aléatoires et $d = 2$:
(distribution uniforme)

$$T = 1.35 \quad [\text{Vallée 97}]$$

Précision : $b + d + \log d - 2$

Algorithme de balayage et cloisonnement

Les prédicats en jeu



● Prédicat 1 : $A_0 <_x A_1$

● Prédicat 3 : $A_0 <_x [A_1A_2] \cap [A_3A_4D]$

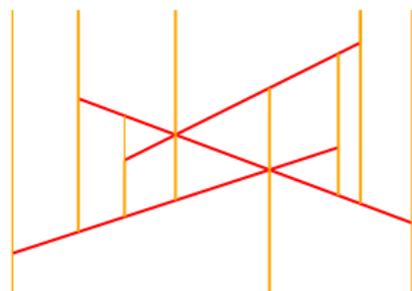
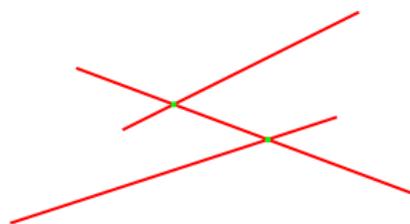
● Prédicat 2 : $A_0 <_y [A_1A_2]$

● Prédicat 4 : $[A_0A_1] \cap [A_2A_3] <_x [A_0A_1] \cap [A_4A_5]$

● Prédicat 2' : $[A_0A_1] \cap [A_2A_3]$

● Prédicat 5 : $[A_0A_1] \cap [A_2A_3] <_x [A_4A_5] \cap [A_6A_7]$

Quel problème veut-on traiter ?



S ensemble de segments \longrightarrow

Pb 1 : paires qui se coupent

Pb 2 : arrangement

Pb 3 : cloisonnement vertical

Prédicat 2' : prédicat d'intersection ($[AB]$, $[CD]$)

Se ramène à 4 prédicats d'orientation

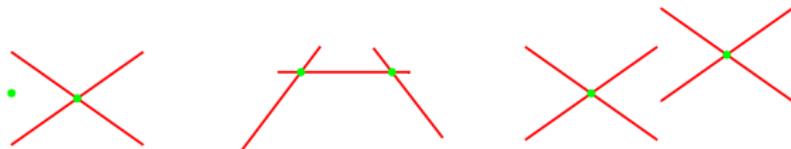
$$(\text{orient}(ABC) \times \text{orient}(ABD) < 0) \wedge (\text{orient}(ACD) \times \text{orient}(BCD) < 0)$$

Degré du prédicat : 2

Prédicats 3 = $A_0 <_x [A_1A_2] \cap [A_3A_4]$

4 : $[A_0A_1] \cap [A_2A_3] <_x [A_0A_1] \cap [A_4A_5]$

5 : $[A_0A_1] \cap [A_2A_3] <_x [A_4A_5] \cap [A_6A_7]$



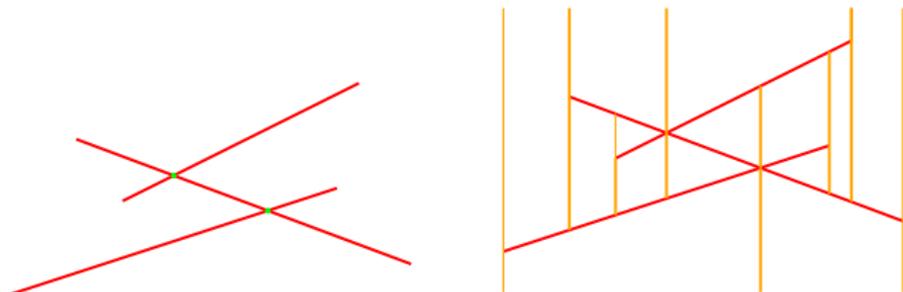
$$x_I = \frac{A_I}{B_I} = x_0 + (x_1 - x_0) \frac{\begin{vmatrix} x_2 - x_0 & y_2 - y_0 \\ x_3 - x_0 & y_3 - y_0 \end{vmatrix}}{\begin{vmatrix} x_1 - x_0 & y_1 - y_0 \\ x_3 - x_2 & y_3 - y_2 \end{vmatrix}}$$

degré du prédicat 3 = 3

degré du prédicat 5 = 5

degré du prédicat 4 = 4 ($A_4A_5 = A_0A_1$)

Des problèmes de degrés différents



Pb 1 (paires sécantes) : 2

Pb 2 (arrangement) : 4

Pb 3 (cloisonnement) : 5

Quelques algorithmes (extraits d'une longue liste !)

Pb1

naive
 $O(n^2)$

Balaban (95)
 $O(n \log n + k)$

Pb2

Chazelle-Edelsbrunner
(92)

$O(n \log n + k)$

Pb3

Bentley-Ottmann (79)
 $O((n + k) \log n)$

RIC (92)
 $O(n \log n + k)$

Quelques algorithmes et leur degré

Pb1

naive
 $O(n^2)$

degré 2

Balaban (95)
 $O(n \log n + k)$

degré 3

Pb2

Chazelle-Edelsbrunner
(92)

$O(n \log n + k)$

degré 4

Pb3

Bentley-Ottmann (79)
 $O((n + k) \log n)$

RIC (92)
 $O(n \log n + k)$

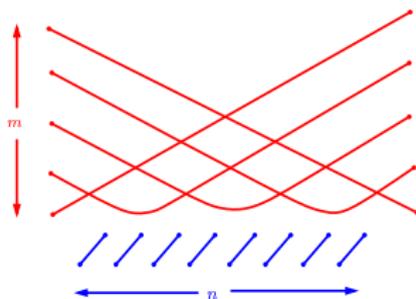
degré 5

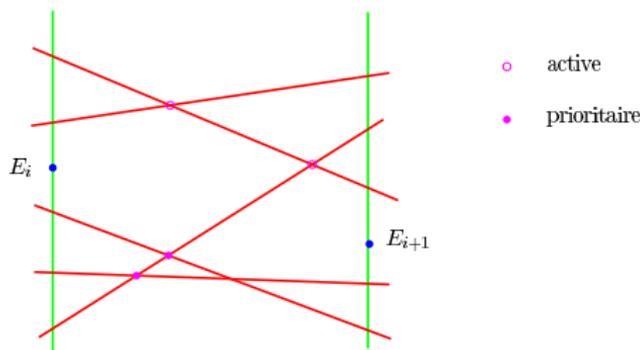
Prédicats et complexité algorithmique

Borne inférieure

La complexité de tout algorithme n'évaluant que les prédicats 1 et 2 est $\Omega(n\sqrt{k})$

[Boissonnat & Snoeyink 98]





Structure de données X :

tableau $E_1 \dots E_{2n}$

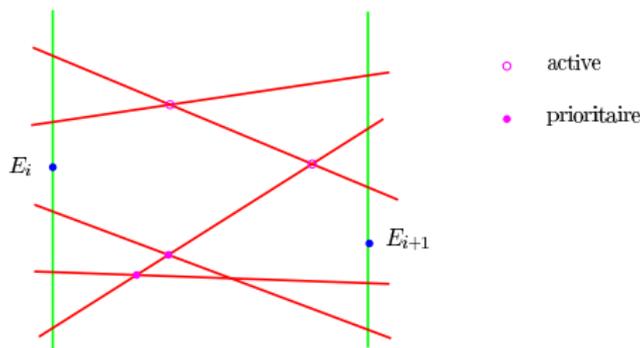
$E_i \longrightarrow$ les paires qui se coupent dans $(E_i, E_{i+1}]$

Algo. :

- Traite les extrémités par $x \nearrow$
- Dans une tranche $(E_i, E_{i+1}]$, traiter les paires adjacentes dans Y dans n'importe quel ordre

Balayage topologique

Prédicats de degré ≤ 3



Préd. 3 suffit pour localiser une intersection dans X

Préd. 2 suffit pour localiser une extrémité dans Y prioritaire

Théorème : L'algorithme de balayage permet de résoudre Pb 1 en temps $O((n + k) \log n)$ en n'utilisant que des prédicats de degré ≤ 3

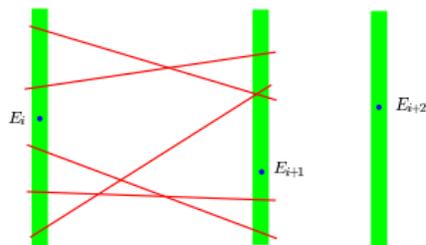
Arrondi exact et double précision

Corollaire : L'algorithme de balayage permet de résoudre Pb 1 en temps $O((n + k) \log n)$ en utilisant de la **double** précision

Prédicat 3 : E extrémité $I = S \cap S'$ $S = [A_0 A_1]$

$$I <_x E \iff \begin{matrix} (x_1 - x_0) \times A < (x_1 - x_E) \times B \\ \text{deg. 2} & \text{deg. 2} \end{matrix}$$

Arrondi exact et monotonie : $I \tilde{<}_x E \implies I <_x E$



$$I \tilde{\in} \text{slab}(E_i, E_{i+1}) \iff E_i \tilde{<}_x I \tilde{<}_x E_{i+2}$$

$$(S, S') \text{ active} \iff E_{i+1} \notin W_I^-$$

Autres résultats

Variante bleu-rouge

Pb1 peut être résolu en temps $O(n \log n + k)$ en n'utilisant des prédicats de degré ≤ 2 [Boissonnat Preparata 1997]

Prédicats de degré ≤ 2 : Pb1 peut être résolu en temps

Arcs : $O(n \log n + \sqrt{k \log \frac{n^2}{k}})$ [Chan 1998]

Segments : $O(n \log^2 n + k \log n)$ [Boissonnat, Snoeyink 1998]

Degré algébrique des prédicats et dimension

Enveloppe convexe in \mathbb{R}^d :

$$\text{orient}(p_0, \dots, p_d) = \text{sign} \begin{vmatrix} p_0 & \dots & p_d \\ 1 & \dots & 1 \end{vmatrix} \text{ de degré } = d$$

Triangulation de Delaunay de points de \mathbb{R}^d :

$$\text{in_sphere}(p_0, \dots, p_d, p_{d+1}) = \text{sign} \begin{vmatrix} p_0 & \dots & p_{d+1} \\ p_0^2 & \dots & p_{d+1}^2 \\ 1 & \dots & 1 \end{vmatrix} \text{ de degré } = d + 2$$

Complexe à témoins (Delaunay faible)

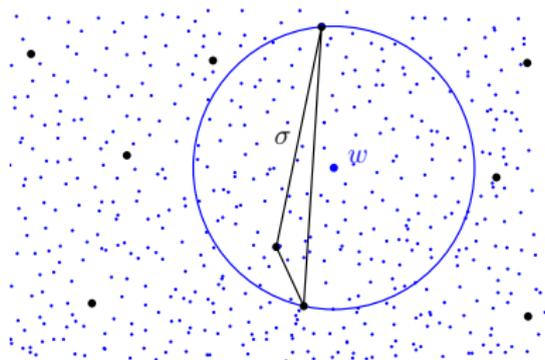
[de Silva]

L un ensemble fini de of points (landmarks)

sommets du complexe

W un échantillon dense (témoins)

pseudo centres



Soit σ un simplexe de sommets dans L , et soit $w \in W$
 w est appelé un **témoins** de σ si

$$\|w - p\| \leq \|w - q\| \quad \forall p \in \sigma \text{ and } \forall q \in L \setminus \sigma$$

Le **complexe à témoin** $\text{Wit}(L, W)$ est l'ensemble de simplexes σ tel que
toute face $\tau \subseteq \sigma$, τ a un témoin dans W

Complexe à témoins (Delaunay faible)

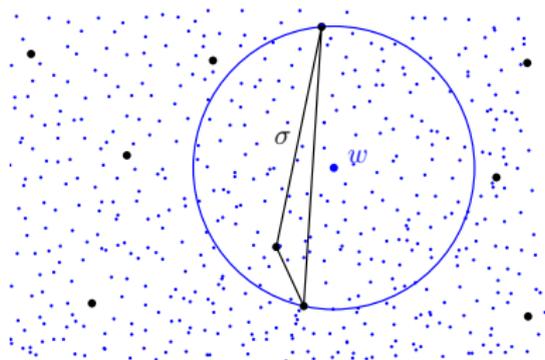
[de Silva]

L un ensemble fini de of points (landmarks)

sommets du complexe

W un échantillon dense (témoins)

pseudo centres



Soit σ un simplexe de sommets dans L , et soit $w \in W$
 w est appelé un **témoin** de σ si

$$\|w - p\| \leq \|w - q\| \quad \forall p \in \sigma \text{ and } \forall q \in L \setminus \sigma$$

Le **complexe à témoin** $\text{Wit}(L, W)$ est l'ensemble de simplexes σ tel que
toute face $\tau \subseteq \sigma$, τ a un témoin dans W

Complexe à témoins (Delaunay faible)

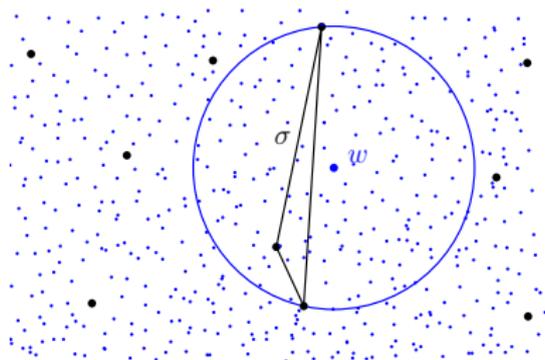
[de Silva]

L un ensemble fini de of points (landmarks)

sommets du complexe

W un échantillon dense (témoins)

pseudo centres



Soit σ un simplexe de sommets dans L , et soit $w \in W$
 w est appelé un **témoins** de σ si

$$\|w - p\| \leq \|w - q\| \quad \forall p \in \sigma \text{ and } \forall q \in L \setminus \sigma$$

Le **complexe à témoin** $\text{Wit}(L, W)$ est l'ensemble de simplexes σ tel que
toute face $\tau \subseteq \sigma$, τ a un témoin dans W

Conséquences de la définition

- Le seul prédicat requis pour construire un complexe à témoins est la comparaison de distances entre paires de points

Distance euclidienne : **degré 2**

- Le complexe à témoins peut être défini dans tout espace métrique et, en particulier, les espaces métriques discrets
- Si $W' \subseteq W$, alors $\text{Wit}(L, W') \subseteq \text{Wit}(L, W)$
- Dans l'espace euclidien : $\text{Del}(L) \subseteq \text{Wit}(L, \mathbb{R}^d)$

Conséquences de la définition

- Le seul prédicat requis pour construire un complexe à témoins est la comparaison de distances entre paires de points

Distance euclidienne : **degré 2**

- Le complexe à témoins peut être défini dans tout espace métrique et, en particulier, les espaces métriques discrets
- Si $W' \subseteq W$, alors $\text{Wit}(L, W') \subseteq \text{Wit}(L, W)$
- Dans l'espace euclidien : $\text{Del}(L) \subseteq \text{Wit}(L, \mathbb{R}^d)$

Conséquences de la définition

- Le seul prédicat requis pour construire un complexe à témoins est la comparaison de distances entre paires de points

Distance euclidienne : **degré 2**

- Le complexe à témoins peut être défini dans tout espace métrique et, en particulier, les espaces métriques discrets
- Si $W' \subseteq W$, alors $\text{Wit}(L, W') \subseteq \text{Wit}(L, W)$
- Dans l'espace euclidien : $\text{Del}(L) \subseteq \text{Wit}(L, \mathbb{R}^d)$

Conséquences de la définition

- Le seul prédicat requis pour construire un complexe à témoins est la comparaison de distances entre paires de points

Distance euclidienne : **degré 2**

- Le complexe à témoins peut être défini dans tout espace métrique et, en particulier, les espaces métriques discrets
- Si $W' \subseteq W$, alors $\text{Wit}(L, W') \subseteq \text{Wit}(L, W)$
- Dans l'espace euclidien : $\text{Del}(L) \subseteq \text{Wit}(L, \mathbb{R}^d)$

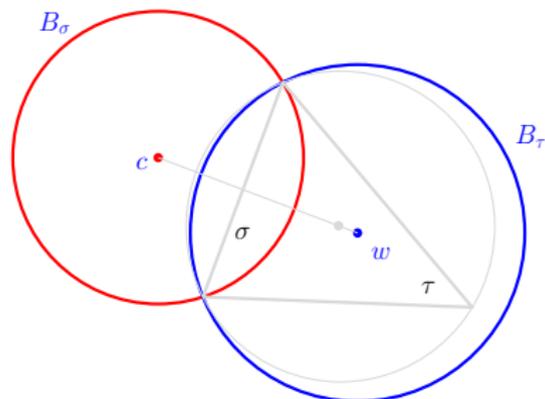
Théorème : $\text{Del}(L) = \text{Wit}(L, \mathbb{R}^d)$

[de Silva 2008]

$\tau = [p_0, \dots, p_k]$ est un k -simplexe de $\text{Wit}(L)$ et B_τ une boule témoin de τ (i.e. $B_\tau \cap L = \tau$)

On montre que $\tau \in \text{Del}(L)$ par induction sur k

Vrai pour $k = 0$



Hyp. : vrai pour $k' \leq k - 1$

$B := B_\tau$

$\sigma := \partial B \cap \tau$, $l := |\sigma|$

// $\sigma \in \text{Del}(L)$ par hyp.

tant que $l + 1 = \dim \sigma < k$ faire

$B \leftarrow$ la boule centrée sur $[cw]$ t. q.

- $\sigma \subset \partial B$,
- B est une boule témoin pour τ
- $|\partial B \cap \tau| = l + 1$

$(B \text{ témoin pour } \tau) \wedge (\tau \subset \partial B) \Rightarrow \tau \in \text{Del}(L)$

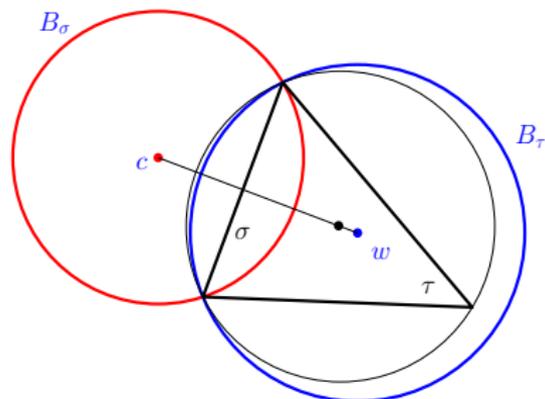
Théorème : $\text{Del}(L) = \text{Wit}(L, \mathbb{R}^d)$

[de Silva 2008]

$\tau = [p_0, \dots, p_k]$ est un k -simplexe de $\text{Wit}(L)$ et B_τ une boule témoin de τ (i.e. $B_\tau \cap L = \tau$)

On montre que $\tau \in \text{Del}(L)$ par induction sur k

Vrai pour $k = 0$



Hyp. : vrai pour $k' \leq k - 1$

$B := B_\tau$

$\sigma := \partial B \cap \tau$, $l := |\sigma|$

// $\sigma \in \text{Del}(L)$ par hyp.

tant que $l + 1 = \dim \sigma < k$ faire

$B \leftarrow$ la boule centrée sur $[cw]$ t. q.

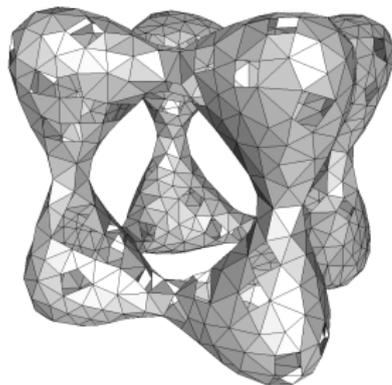
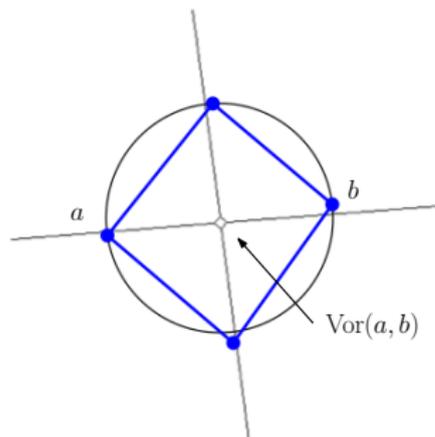
- $\sigma \subset \partial B$,
- B est une boule témoin pour τ
- $|\partial B \cap \tau| = l + 1$

$(B \text{ témoin pour } \tau) \wedge (\tau \subset \partial B) \Rightarrow \tau \in \text{Del}(L)$

Cas d'un ensemble fini de points W :

$$\text{Wit}(L, W) \neq \text{Del}(L)$$

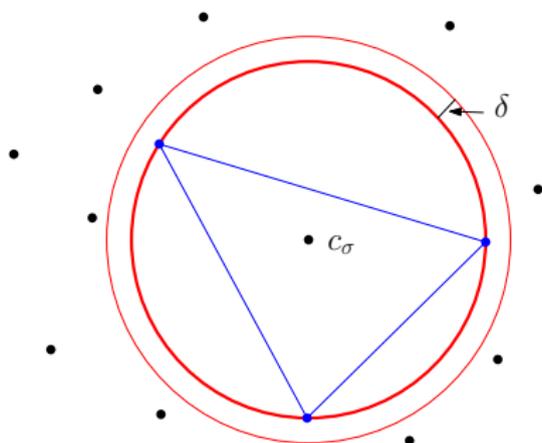
$\text{Wit}(L, W) \neq \text{Del}(L)$, même si W est un échantillon dense de Ω



$$[ab] \in \text{Wit}(L, W) \Leftrightarrow \exists p \in W, \text{Vor}_2(a, b) \cap W \neq \emptyset$$

Triangulations de Delaunay protégées

Quantifier la généricité



On dit qu'un simplexe de Delaunay $\sigma \subset L$ est δ -protégé si

$$\|c_\sigma - q\| > \|c_\sigma - p\| + \delta \quad \forall p \in \sigma \text{ et } \forall q \in L \setminus \sigma.$$

Protection implique égalité : $\text{Wit}(L, W) = \text{Del}(L)$

Lemma

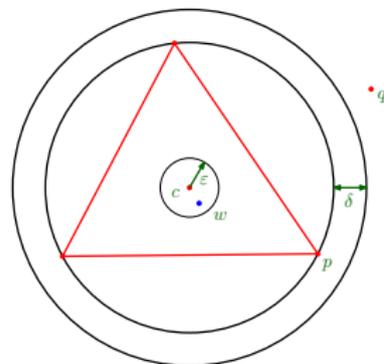
W et L deux ensembles finis de points de \mathbb{T}^d

W est ε -dense

$$\Rightarrow \text{Wit}(L, W) = \text{Del}(L)$$

$\forall \sigma \in \text{Del}(L)$, σ est δ -protégé avec $\delta \geq 2\varepsilon$,

Démonstration Il suffit de montrer $\text{Wit}(L, W) \supseteq \text{Del}(L)$



$$\forall p \in \sigma, \forall q \in L \setminus \sigma, \exists w \in W \cap B(c, \varepsilon)$$

$$\begin{aligned} \|w - p\| &\leq \|w - c\| + \|c - p\| \\ &\leq \|w - c\| + (\|c - q\| - \delta) \\ &\leq \|w - q\| + 2\|w - c\| - \delta \\ &\leq \|w - q\| + 2\varepsilon - \delta \end{aligned}$$

Si $\delta \geq 2\varepsilon$, w est donc un témoin de σ

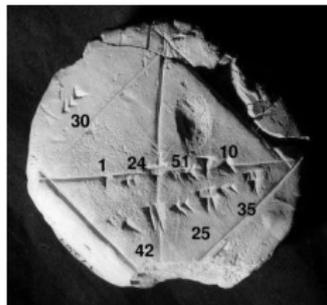
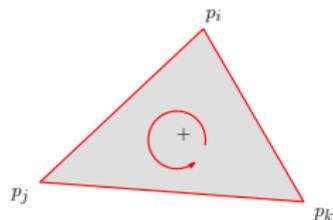
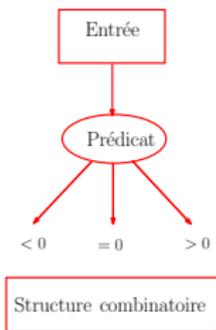
□

Algorithme

- 1 Calculer $\text{Wit}(L, W)$
- 2 Si $\text{Wit}(L, W)$ n'est pas une triangulation (trous), choisir une des stratégies suivantes :
 - ▶ Raffiner W
 - ▶ Perturber ou raffiner ou pondérer L

Analyse : cours suivant

Interactions entre géométrie et calcul

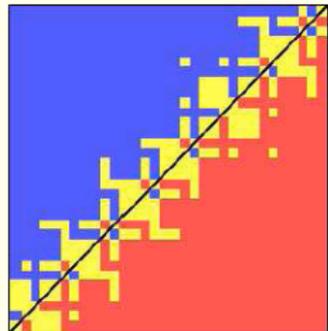


$$p = (0.5 + x.u, 0.5 + y.u)$$
$$0 \leq x, y < 256, u = 2^{-53}$$
$$q = (12, 12)$$
$$r = (24, 24)$$

orientation(p, q, r)
evaluated with double

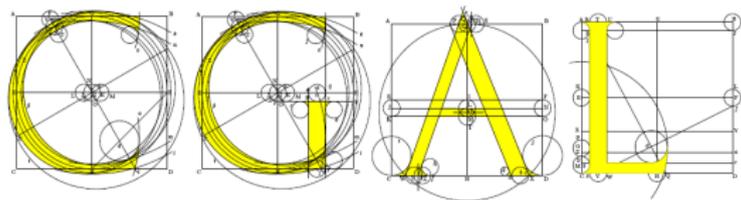
256 x 256 pixel image

> 0, = 0, < 0



Kettner et al. [2008]

Une bibliothèque logicielle pour la géométrie algorithmique



Computational Geometry Algorithms Library

