

# Géométrie Algorithmique

## Triangulation de Delaunay

- Algorithmes de construction d'une triangulation de Delaunay

Deux approches

1- Construction d'une triangulation pour un ensemble de points donnés

- Construction du diagramme de Voronoï puis « dualisation »
- Algorithme ad-hoc incrémentaux ou non

2- Les points sont calculés au fur et à mesure de la construction de la triangulation

- Génération de maillage / adaptation de maillage
- Algorithme nécessairement incrémentaux

# Géométrie Algorithmique

## Triangulation de Delaunay

- Principe des algorithmes incrémentaux de construction d'une triangulation de Delaunay
  - Insertion d'un point  $p_n$  dans une triangulation à  $n-1$  points
    - Trouver l'entité contenant le nouveau point
    - Insérer le point
    - Modifier la triangulation pour garder le caractère « de Delaunay » de celle ci

Il existe deux algorithmes voisins (équivalents puisque la triangulation de Delaunay est unique)
  - Lawson – edge swapping

Garantit une triangulation valide à toute étape de l'algorithme.
  - Bowyer-Watson – trouver tous les triangles dont le centre circonscrit contient le nouveau point, détruire ces triangles et reconstruire la cavité.

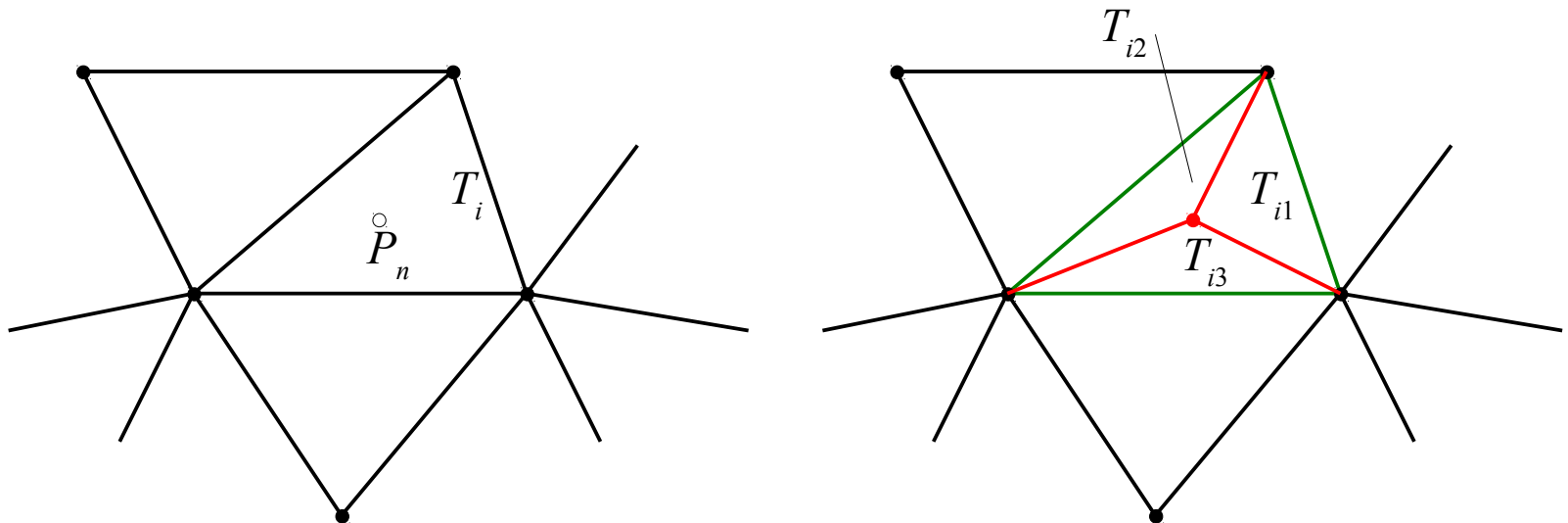
Inconvénient : possible de construire des cavités non étoilées (voire non connexes) si les calculs sont faits en précision finie.

# Géométrie Algorithmique

## Triangulation de Delaunay

- Recherche de l'entité contenant le point – cf suite du cours (localisation)
- Insertion du point dans la triangulation

Cas 1 : le point  $p_n$  est situé dans un triangle  $T_i$



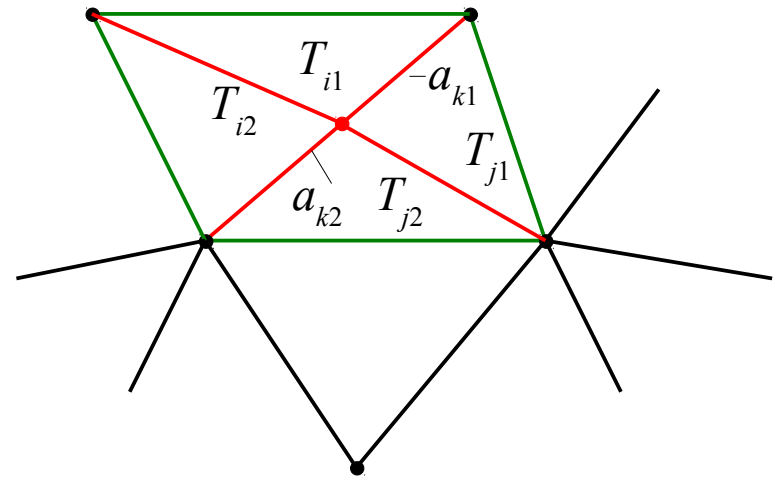
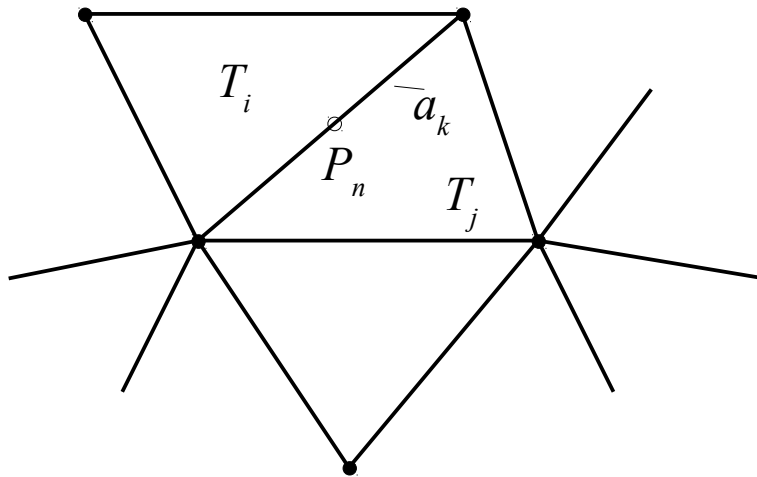
- Les nouvelles arêtes **rouges** sont nécessairement légales.

# Géométrie Algorithmique

## Triangulation de Delaunay

- Insertion du point dans la triangulation

Cas 2 : le point  $p_n$  est situé sur une arête  $a_k$

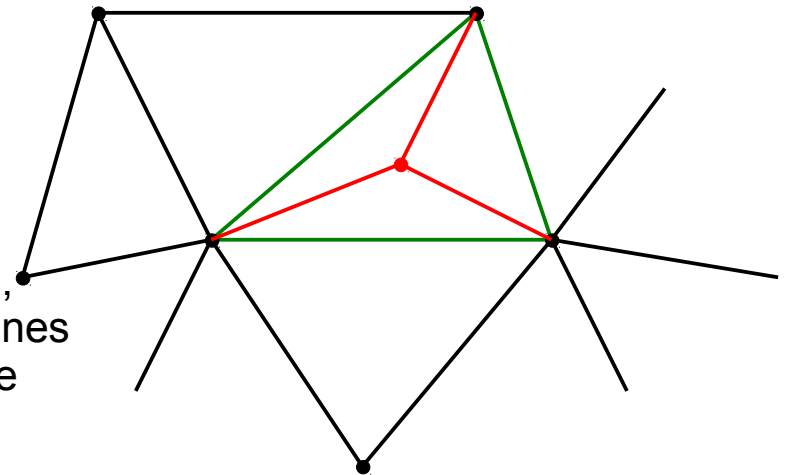


- Les nouvelles arêtes **rouges** sont nécessairement légales.
- Qu'en est il des autres (en particulier les **vertes**) ?

# Géométrie Algorithmique

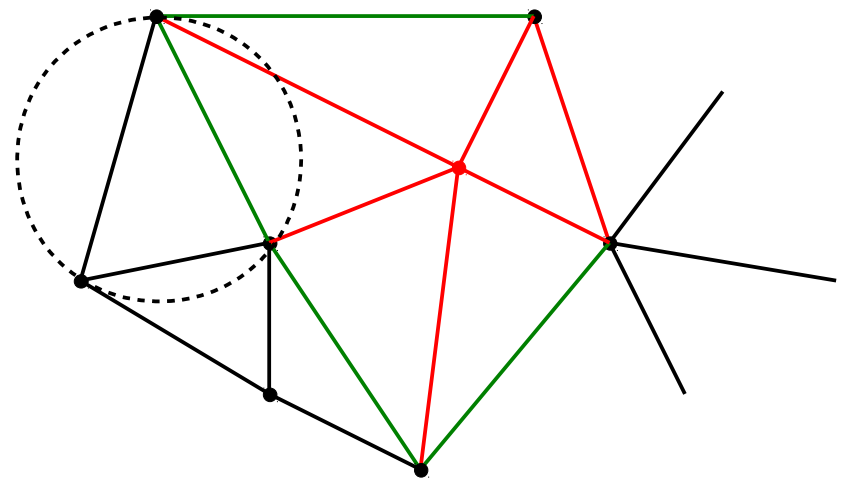
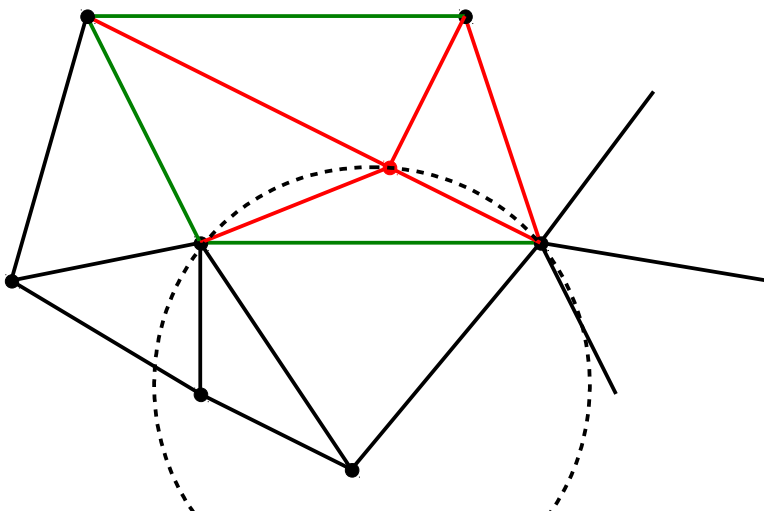
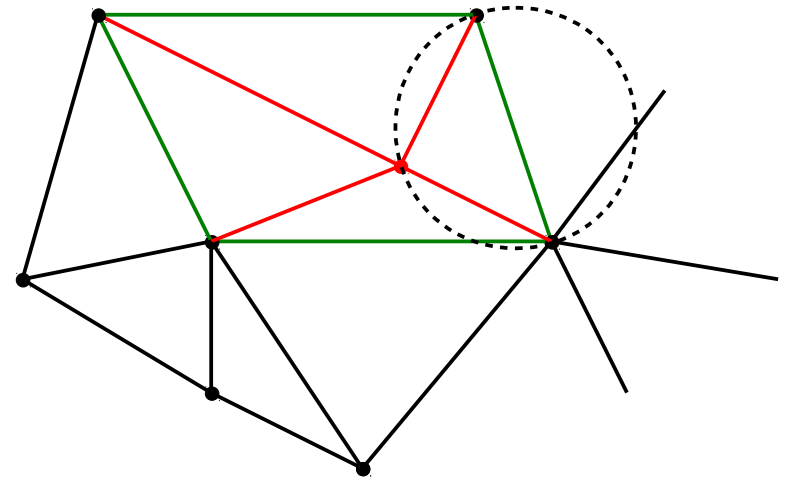
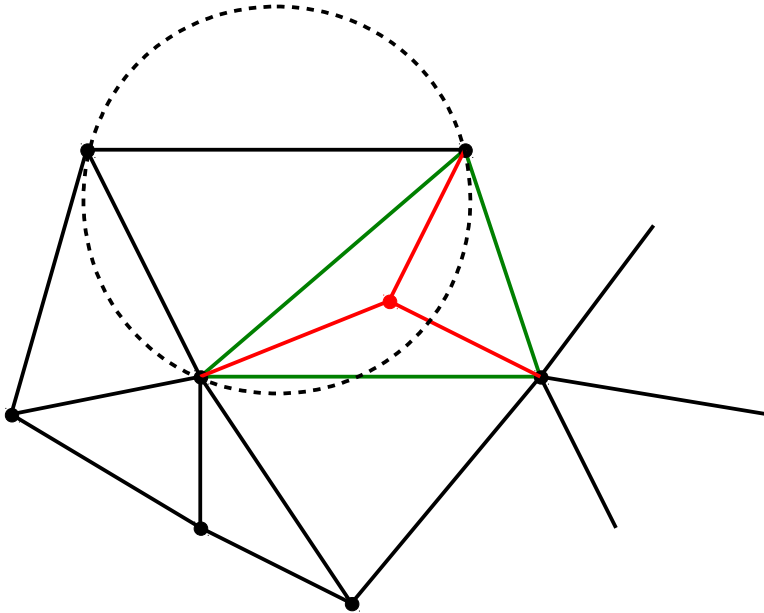
## Triangulation de Delaunay

- Modifier la triangulation pour garder le caractère « de Delaunay » de celle ci
  - Revient à légaliser toute arête qui devient illégale suite à l'insertion du nouveau sommet dans la triangulation
  - Quelles sont ces arêtes ?
    - Ce sont potentiellement toutes celles pour lesquelles de nouveaux triangles voisins sont apparus (ici en **vert**)
    - Toutes les autres arêtes (noires) sont pour le moment légales, car leurs voisins sont identiques.
    - Il faut donc tester ces arêtes vertes pour voir si elles sont légales, et les légaliser le cas échéant par retournements d'arêtes.
    - Lors d'un retournement d'arête, il faut vérifier si les arêtes voisines d'un triangle détruit sont encore légales ...



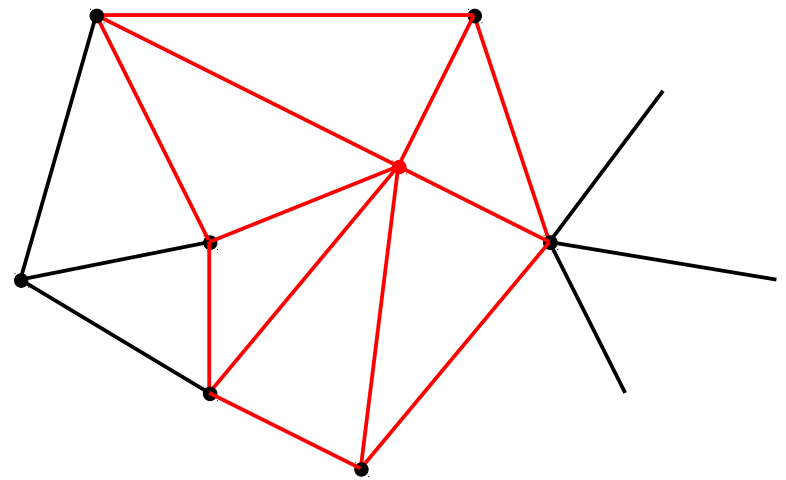
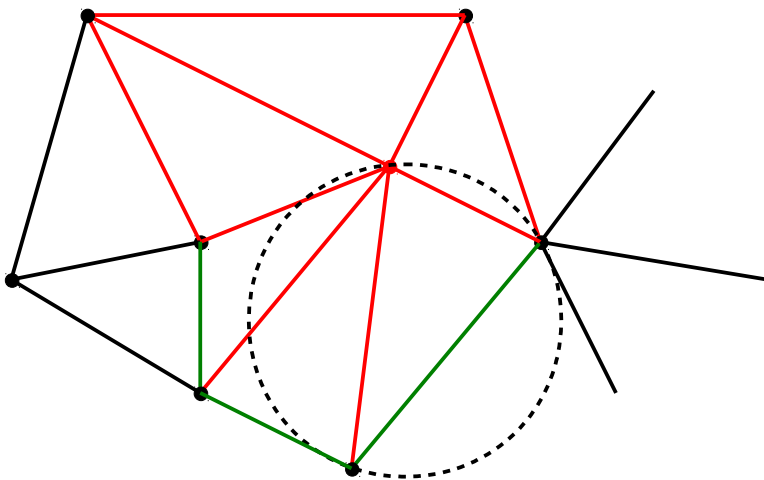
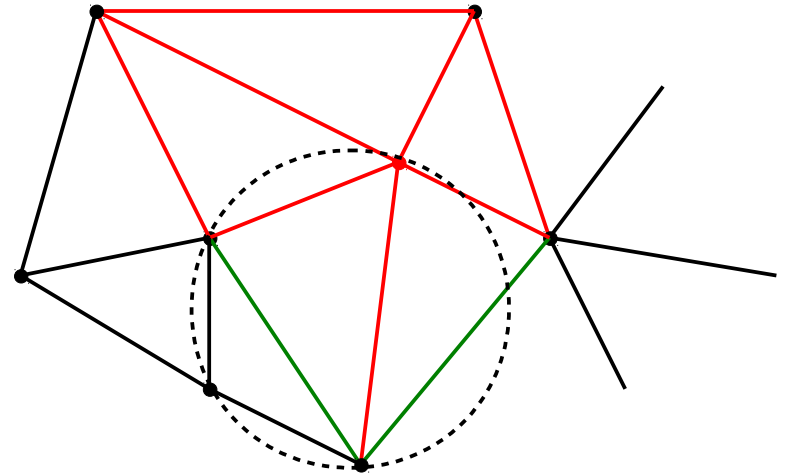
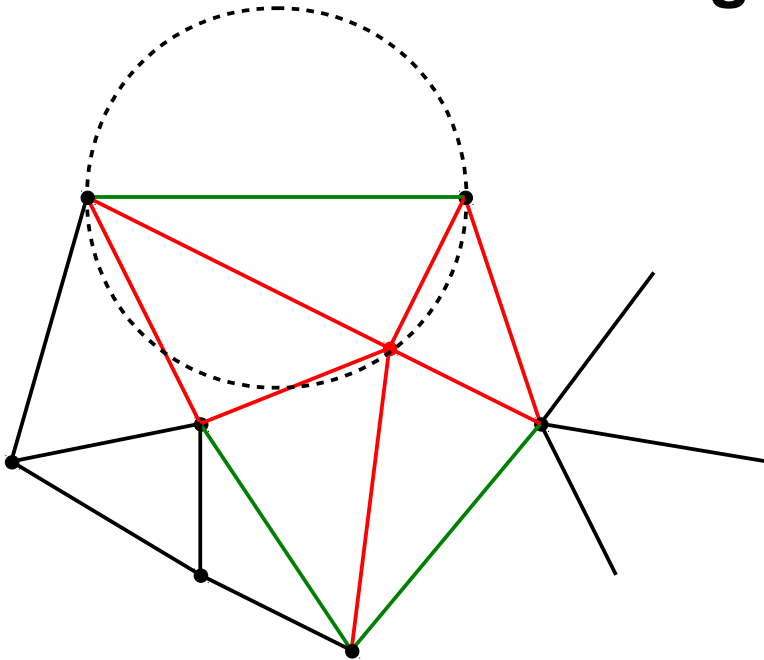
# Géométrie Algorithmique

## Triangulation de Delaunay



# Géométrie Algorithmique

## Triangulation de Delaunay



# Géométrie Algorithmique

## Triangulation de Delaunay

- Algorithme global

**InsererPoint( $p_r, T$ )**

**Entree :** un point  $p_r$  à inserer et une triangulation de Delaunay

**Sortie :** triangulation de Delaunay contenant  $p_r$ .

{

    Trouver le triangle  $T_i (p_i, p_j, p_k)$  contenant  $p_r$

    Si  $p_r$  est à l'intérieur de  $T_i$

    {

        Couper  $T_i$  en 3 et ajouter les arêtes entre  $p_r$  et  $p_i, p_j$  et  $p_k$

        LegaliseArete( $p_r, p_i, T$ )

        LegaliseArete( $p_r, p_j, T$ )

        LegaliseArete( $p_r, p_k, T$ )

    }

    Sinon ( $p_r$  est sur une arete ,  $p_i p_j$  par exemple,  $p_k$  et  $p_l$  sont les deux points opposes à  $p_i p_j$ )

    {

        Couper les deux triangles voisins de  $p_i p_j$  en 4 et inserer les aretes entre  $p_r$  d'une part et  $p_i, p_j, p_k, p_l$  d'autre part

        LegaliseArete( $p_r, p_i, T$ ) ;   LegaliseArete( $p_r, p_j, T$ ) ;

        LegaliseArete( $p_r, p_k, T$ ) ;   LegaliseArete( $p_r, p_l, T$ ) ;

    }

}



# Géométrie Algorithmique

## Triangulation de Delaunay

- Légalisation d'une arête

**LegaliseArete( $p_i, p_j, T$ )**

{

Si  $(p_i p_j)$  est une arête illégale

{

Soit  $p_i p_j p_k$  le triangle adjacent à  $p_i p_j$  au travers de l'arête  $p_i p_j$

Retourner l'arête  $p_i p_j$  en la remplaçant par  $p_i p_k$

LegaliseArete( $p_i, p_k, T$ )

LegaliseArete( $p_i, p_k p_j, T$ )

}

}

# Géométrie Algorithmique

## Triangulation de Delaunay

- Preuve que l'algorithme est correct

Il faut prouver qu'aucune arête illégale ne reste après l'insertion du sommet  $p_r$ .

- On peut noter que :

- Chaque nouvelle arête créée est reliée à  $p_r$
- Une arête ne peut devenir illégale que lorsque un triangle voisin est modifié (par l'insertion de  $p_r$  ou par retournement d'arête) – ainsi elle sera traitée ultérieurement lors de la récursion
- Chaque nouvelle arête créée est nécessairement légale et fait partie du graphe de Delaunay (preuve ci – après)

CQFD.

# Géométrie Algorithmique

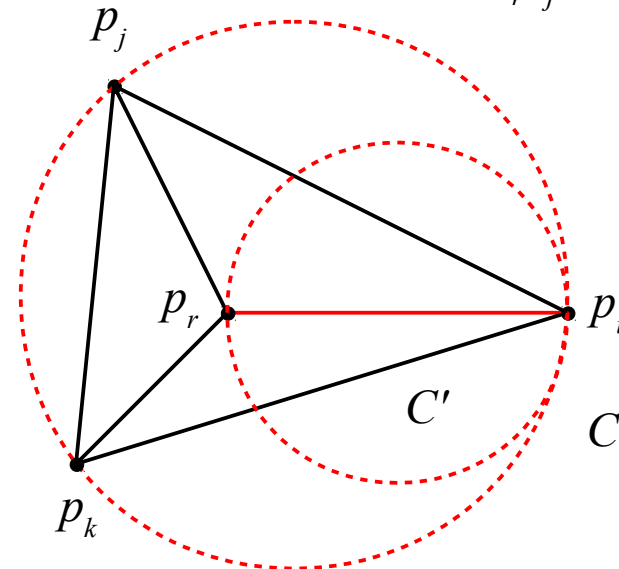
## Triangulation de Delaunay

- Les nouvelles arêtes appartiennent à la triangulation de Delaunay

Cas de l'insertion du point  $p_r$ .

Considérer les arêtes  $p_i p_j$ ,  $p_i p_r$  et  $p_j p_r$ . Comme  $p_i p_j p_k$  est un triangle datant d'avant l'insertion de  $p_r$ , son cercle circonscrit  $C$  ne contient aucun autre point que  $p_r$ . En diminuant la taille de  $C$ , on peut faire passer un cercle  $C'$  par  $p_r$  et  $p_i$  contenu dans  $C$ . Il ne contient alors aucun point. Cela implique que  $p_i p_r$  fait partie de la triangulation de Delaunay (et par permutation il en est de même avec  $p_j p_r$  et  $p_r p_k$ ).

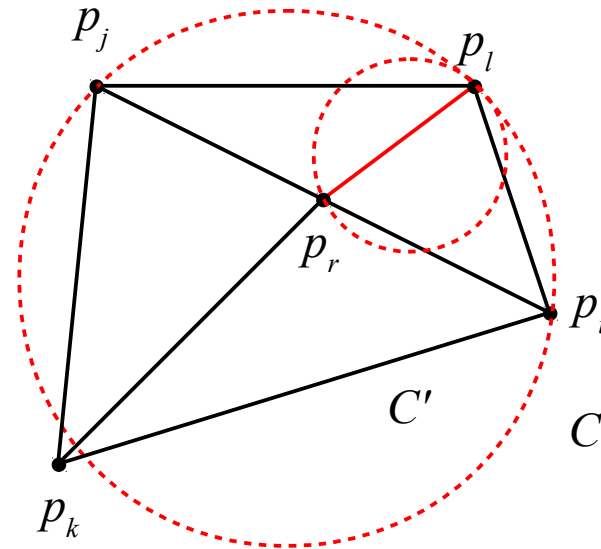
- Preuve identique si  $p_r$  est sur une arête.



# Géométrie Algorithmique

## Triangulation de Delaunay

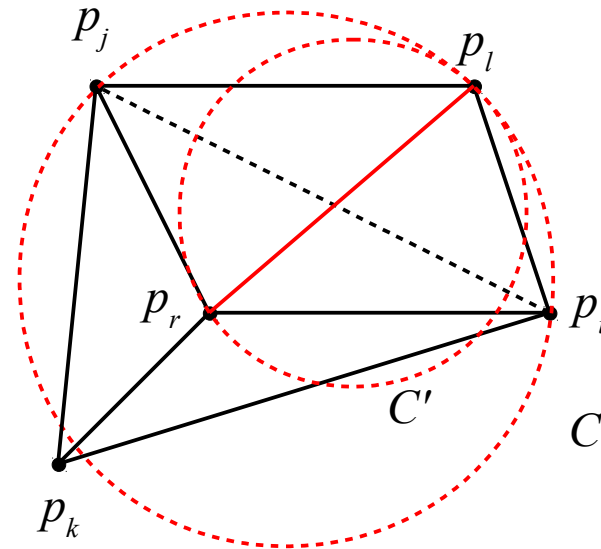
- Preuve identique si  $p_r$  est sur une arête.



# Géométrie Algorithmique

## Triangulation de Delaunay

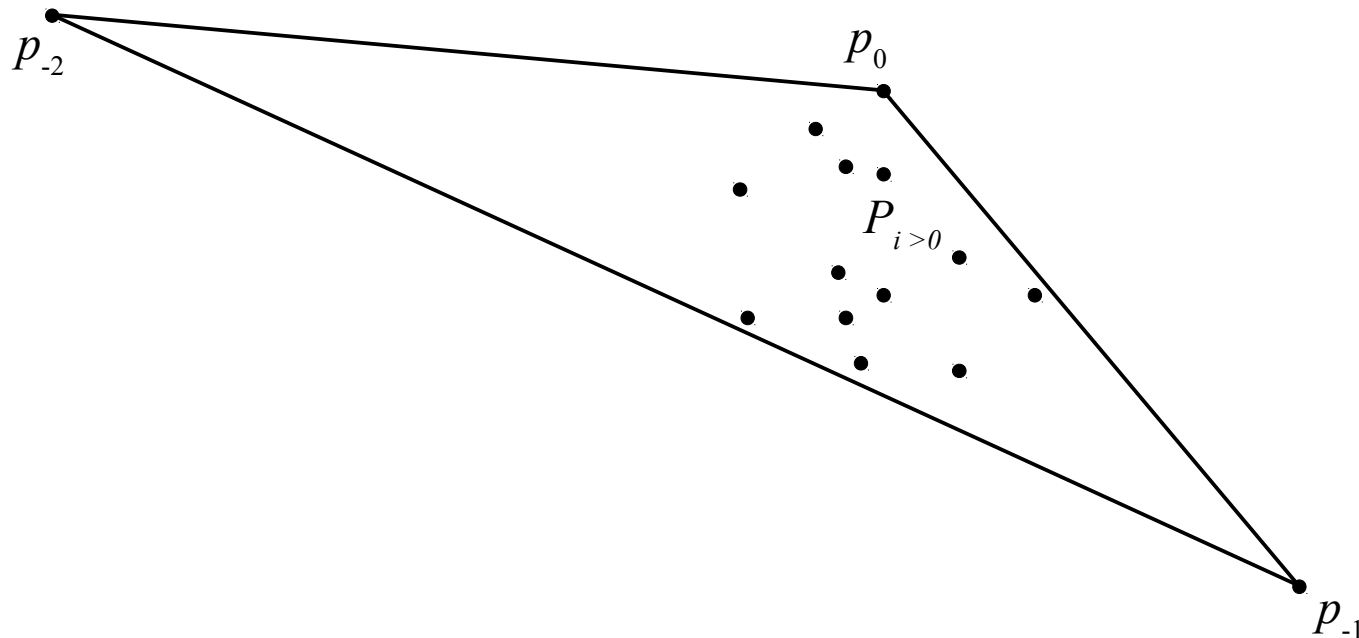
- Les nouvelles arêtes appartiennent à la triangulation de Delaunay  
Cas du retournement d'une arête suite à l'insertion de  $p_r$   
On remplace une arête  $p_i p_j$  par une arête  $p_i p_r$ . Comme  $p_i p_j p_l$  est un triangle datant d'avant l'ajout de  $p_r$ , son cercle circonscrit  $C$  ne contient que  $p_r$  (et si il ne contenait pas  $p_r$ , alors  $p_i p_j$  serait légale). Il est alors possible de trouver un autre cercle  $C'$  passant par  $p_r$  et  $p_l$ , contenu dans  $C$ , donc ne contenant aucun point. Ceci prouve que  $p_i p_r$  fait bien partie de la triangulation de Delaunay.



# Géométrie Algorithmique

## Triangulation de Delaunay

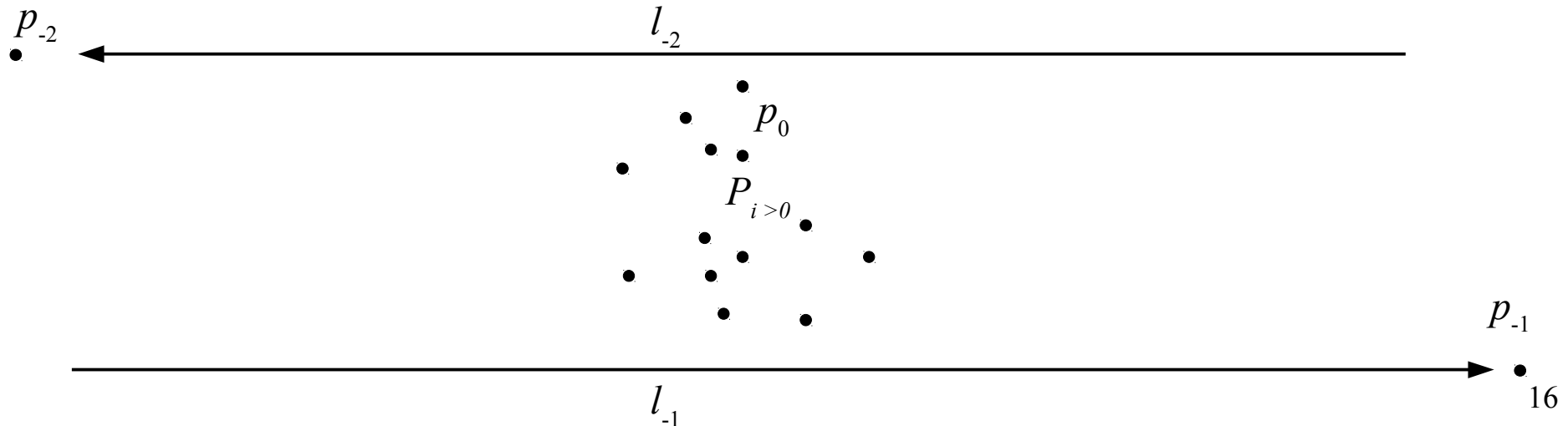
- Qu'en est il de l'initialisation ?
  - On part d'une triangulation triviale comprenant un des sommets situés sur l'enveloppe convexe et deux sommets fictifs
    - Ils sont situés loin des autres points (en particulier, ils ne doivent être contenus dans aucun cercle circonscrit à des triangles de la triangulation finale)
    - On ne peut les éloigner arbitrairement (pour des raisons numériques)



# Géométrie Algorithmique

## Triangulation de Delaunay

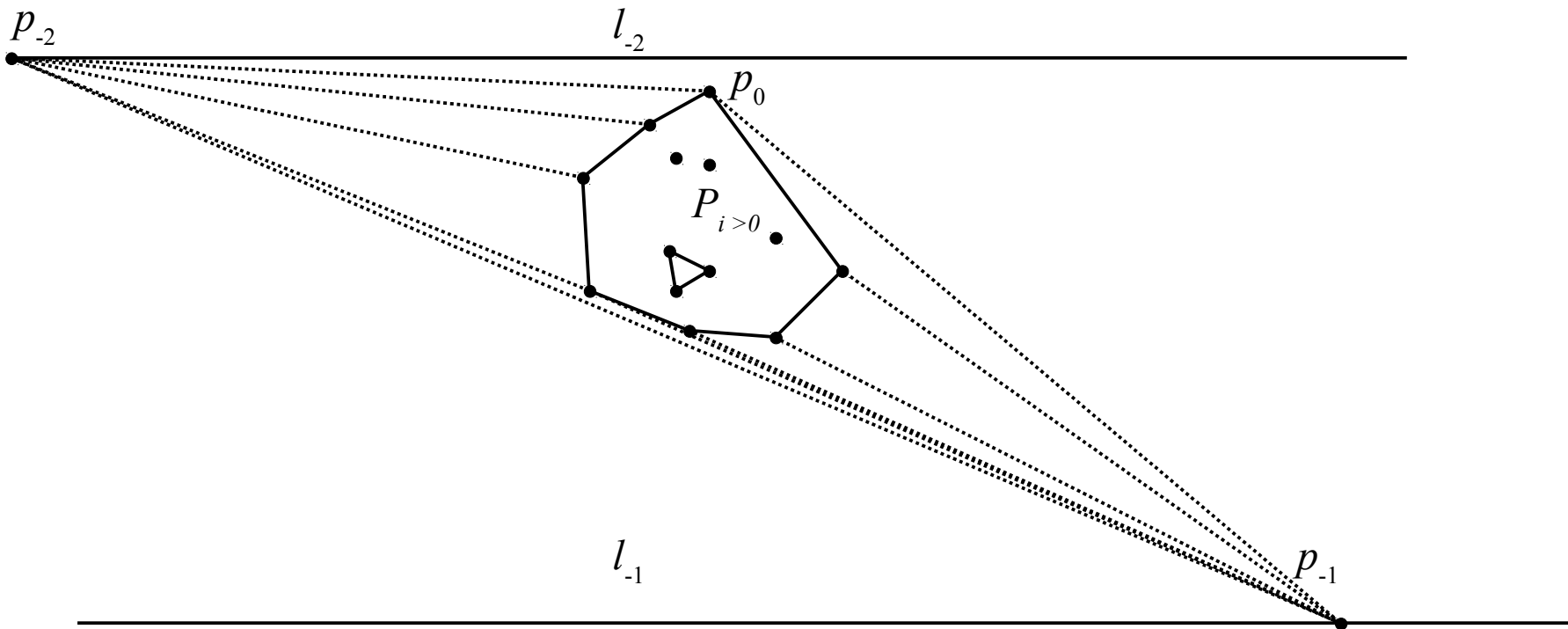
- On ne va en fait PAS assigner de coordonnées à  $p_{-1}$  et  $p_{-2}$ , mais modifier tous les tests impliquant ces points de façon à ce qu'ils se comportent comme si ils étaient situés à l'infini.
  - Soit le classement suivant :  $p$  est au dessus de  $q$  si  $y_p > y_q$  ou si  $y_p = y_q$  et  $x_q > x_p$ .
  - Soient  $l_{-1}$  et  $l_{-2}$  deux lignes (cf dessin).  $p_{-1}$  est sur  $l_{-1}$  et tel que l'ordre induit par le classement est le même que l'ordre selon un classement dans le sens horaire autour de  $p_{-1}$ .  $p_{-2}$  est sur  $l_{-2}$  et tel que l'ordre induit autour de  $p_{-2}$  est antihoraire – pour tous les points  $p_i$  ET  $p_{-1}$



# Géométrie Algorithmique

## Triangulation de Delaunay

- La triangulation de Delaunay de  $\{p_{-2}, p_{-1}, p_0 \dots p_n\}$  est la triangulation de  $\{p_0 \dots p_n\}$  avec les arêtes joignant la partie droite de l'enveloppe convexe à  $p_{-1}$  et la partie gauche à  $p_{-2}$  ainsi que  $p_{-2}p_{-1}$ .



- Une fois la triangulation construite, il suffit de retirer tous les triangles connectés à  $p_{-2}$  et  $p_{-1}$ .



# Géométrie Algorithmique

## Triangulation de Delaunay

- Modification du test de légalité
  - Soit  $p_i p_j$  une arête dont on veut savoir si elle est légale. Les points situés de part et d'autre de cette arête sont  $p_k$  et  $p_l$ .
    - 1 - Si  $p_i p_j$  est une arête du triangle  $p_0 p_{-1} p_{-2}$  alors elle est légale.
    - 2 - Les indices  $i, j, k, l$  sont tous non négatifs : cas classique
    - 3 - Dans les autres cas,  $p_i p_j$  est légale ssi  $\min(k, l) < \min(i, j)$ .Explication :
    - 3a - Si un seul de  $i, j, k, l$  est négatif, alors l'arête joignant deux points d'indices positifs est la seule légale, soit  $ij$ , soit  $kl$ .
    - 3b - Si deux indices de  $i, j, k, l$  sont négatifs, alors nécessairement ceux ci sont répartis entre  $(i, j)$  et  $(k, l)$  (sinon on est dans le cas 1, noter que dans tous les cas l'indice  $r > 0$  du point inséré fait partie de  $i, j, k, l$ )  
Alors, l'arête légale est celle contenant  $p_{-1}$  (logique car on a décidé que  $p_{-2}$  était placé de telle façon que aucun cercle circonscrit de la triangulation ne le contienne, y compris des triangles formés avec  $p_{-1}$ )

# Géométrie Algorithmique

## Triangulation de Delaunay

- Performance de l'algorithme est en  $n \log n$  (Optimal)
  - A condition que la performance de la recherche du triangle contenant un point soit en  $\log n$  (cf cours prochain)

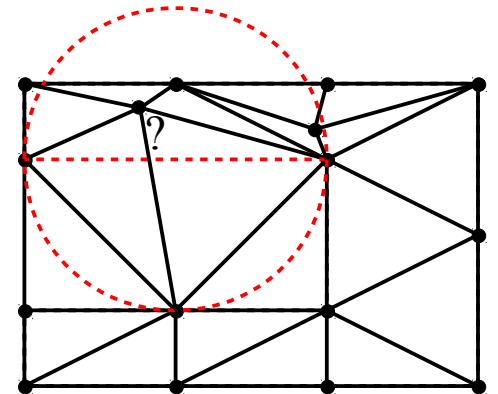
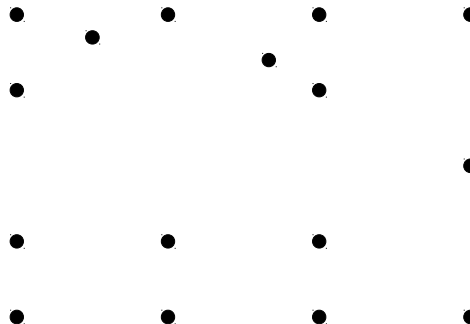
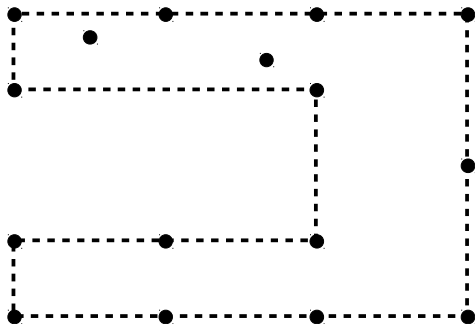
# Géométrie Algorithmique

## Triangulation de Delaunay

- L'algorithme précédent permet de trianguler l'enveloppe convexe d'un ensemble de points.

On dispose souvent d'une discrétisation des contours d'un polygone (possiblement non convexe, et possiblement troué)

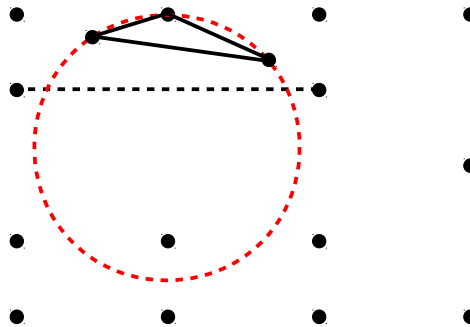
- On doit alors faire apparaître exceptionnellement certaines arêtes dans la triangulation finale, et celles ci ne sont pas forcément légales.



# Géométrie Algorithmique

## Triangulation de Delaunay

- Une triangulation dans laquelle certaines arêtes sont forcées est une triangulation contrainte.
  - Si elle respecte le critère « contraint » de la sphère vide, en ce sens que la sphère circonscrite à tout triangle  $p_i p_j p_k$  ne contient aucun point visible depuis  $p_i$ ,  $p_j$  et  $p_k$ , alors c'est une Triangulation de Delaunay Contrainte.



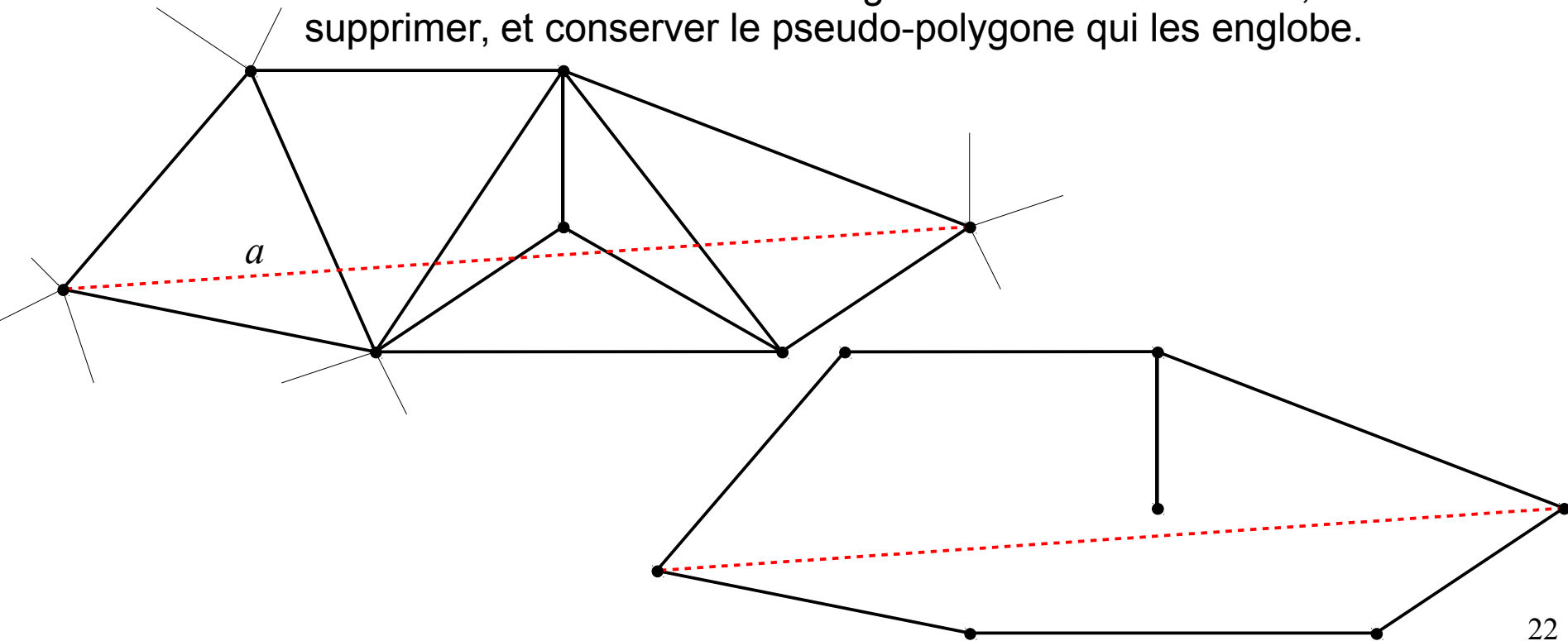
- Définition de la visibilité : deux points  $a$  et  $b$  sont visibles si le segment  $ab$  n'intersecte aucune arête contrainte.

# Géométrie Algorithmique

## Triangulation de Delaunay

- Comment imposer à une triangulation de Delaunay de respecter certaines arêtes ?
  - Algorithme :

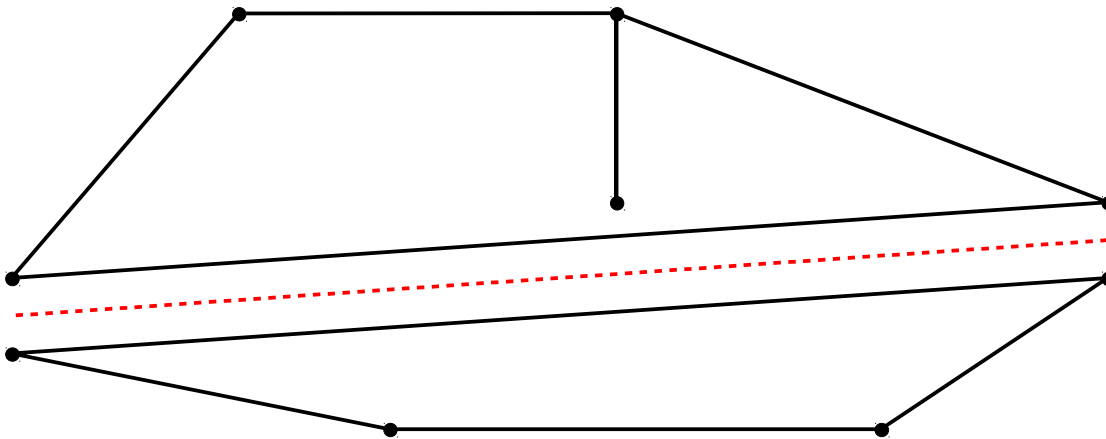
Déterminer l'ensemble des triangles intersectant l'arête  $a$ , les supprimer, et conserver le pseudo-polygone qui les englobe.



# Géométrie Algorithmique

## Triangulation de Delaunay

- Couper le pseudo-polygone en deux parties par l'arête  $a$ . Celle ci fait nécessairement partie de l'enveloppe convexe des deux parties.

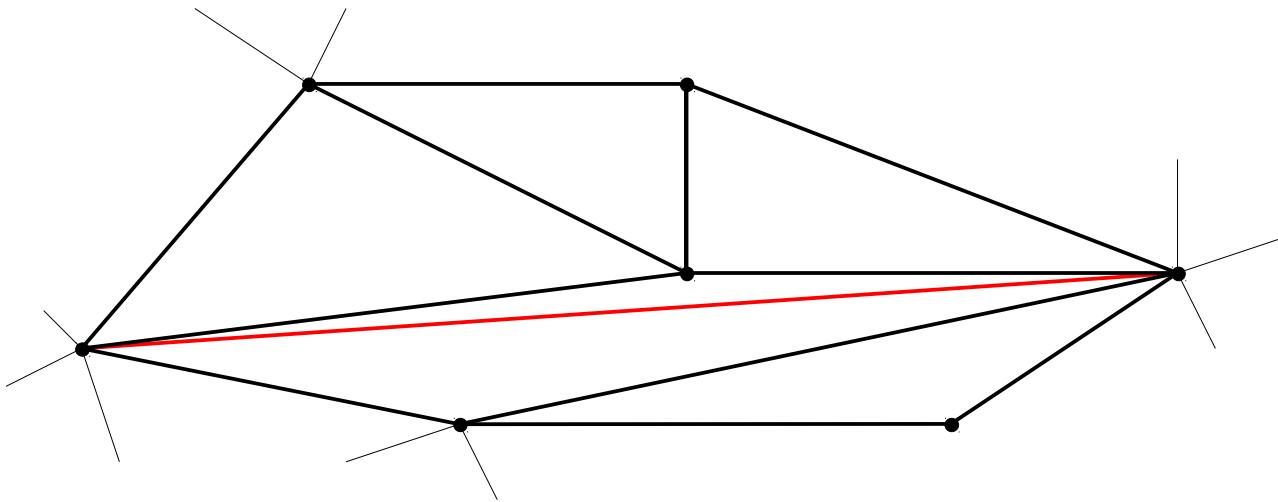


- Noter que les arêtes des deux polygones sont forcément légales et font partie de la triangulation de Delaunay des sommets.
  - Preuve : elles étaient légales et l'on a retiré des sommets, elles restent donc légales. De plus, l'arête  $a$  fait nécessairement partie elle aussi de la triangulation de Delaunay car elle est sur l'enveloppe convexe des sommets.

# Géométrie Algorithmique

## Triangulation de Delaunay

- Triangler chaque pseudo-polygone séparément et rassembler cela dans la triangulation de départ.



- Noter que les pseudo-polygones ne sont pas convexes et qu'il se peut que des triangles « extérieurs » apparaissent. Ceux ci doivent être éliminés.

# Géométrie Algorithmique

## Triangulation de Delaunay

- Génération de maillage

- On suppose connue une triangulation de Delaunay d'un contour donné

Le but ici est de générer des sommets supplémentaires afin que la triangulation respecte certains critères

- Taille des triangles
- Aspect de forme
- Où insérer ces points supplémentaires dans la triangulation ?
  - Idée : profiter du fait que l'on dispose à l'étape  $n-1$  d'une triangulation contrainte valide.
  - On peut mesurer un critère selon les triangles ou les arêtes de la triangulation, et insérer un sommet judicieusement afin d'améliorer localement ce critère
  - Il faut refaire la triangulation du voisinage à l'aide de l'algorithme de retournement d'arête de Lawson à chaque insertion.
  - Il existe des arête inviolables (contours imposés) donc on doit modifier l'algorithme de Lawson de façon à interdire l'inversion de ces arêtes.



# Géométrie Algorithmique

## Triangulation de Delaunay

- Légalisation d'une arête (avec contraintes)

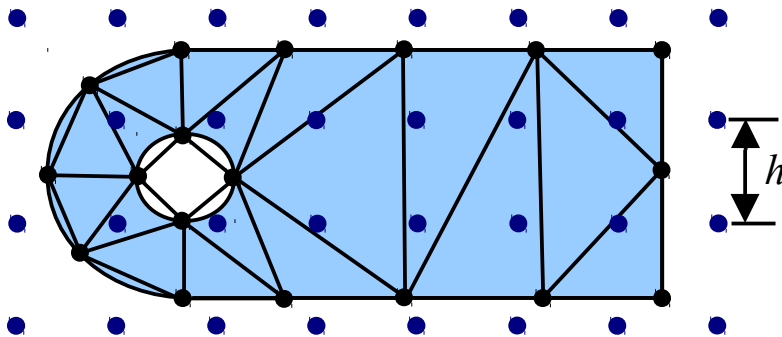
**LegaliseArete( $p_i, p_j, T$ )**

```
{  
  Si ( $p_i p_j$ ) est une arête illégale & qu'elle n'est pas une contrainte  
  {  
    Soit  $p_i p_k p_j$  le triangle adjacent à  $p_i p_j$  au travers de l'arête  $p_i p_j$   
    Retourner l'arete  $p_i p_j$  en la remplaçant par  $p_i p_k$   
    LegaliseArete( $p_i, p_k, T$ )  
    LegaliseArete( $p_i, p_k p_j, T$ )  
  }  
}
```

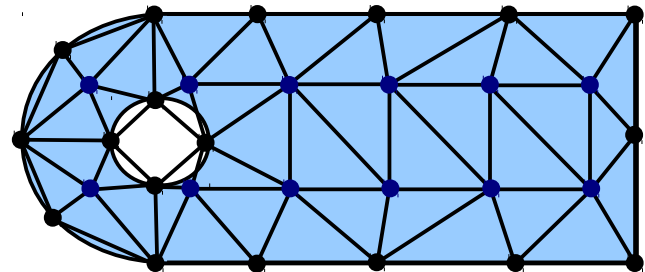
# Géométrie Algorithmique

## Triangulation de Delaunay

- Stratégies de placement de nouveaux sommets
  - Grille régulière



- Les sommets extérieurs sont ignorés

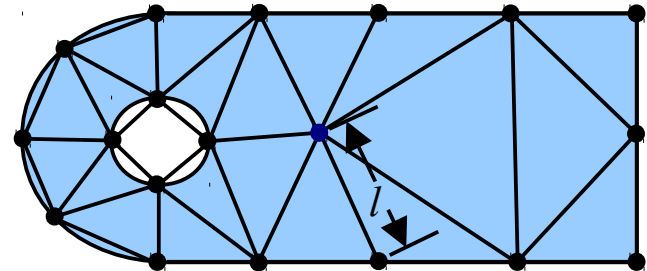
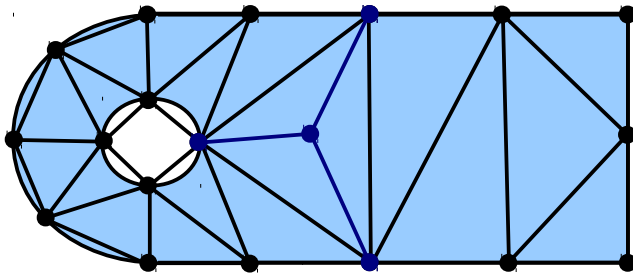


# Géométrie Algorithmique

## Triangulation de Delaunay

- Centroïde

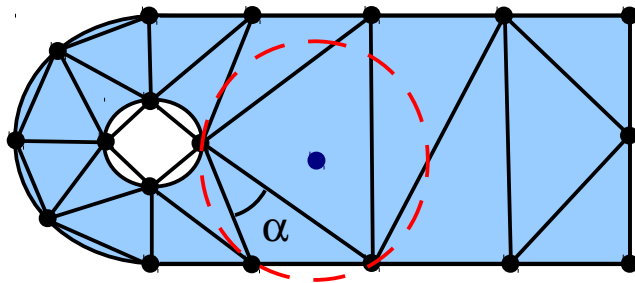
On insère des sommets jusqu'à ce que les longueurs d'arêtes soient convenables



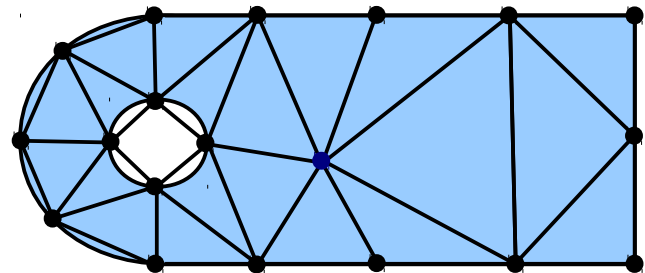
# Géométrie Algorithmique

## Triangulation de Delaunay

- Centre du cercle circonscrit (P. Chew, J. Ruppert, J. Shewchuk)  
l'ordre d'insertion est basé sur l'angle minimal d'un triangle, on insère jusqu'à ce que l'angle minimal soit supérieur à une valeur (en principe  $30^\circ$ )



- On peut prouver que cela se termine pour tout angle inférieur à  $20.7^\circ$  (cf J. Ruppert, 1995)  
en pratique cela marche jusque  $\sim 33.8^\circ$  (cf J. Shewchuk, 1996)

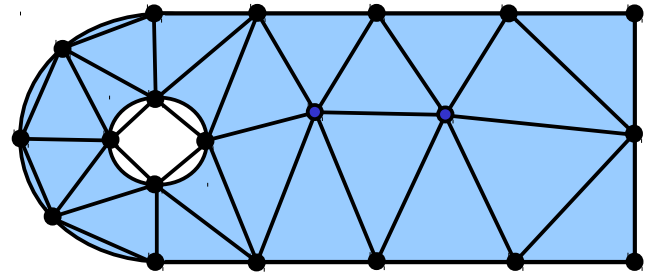
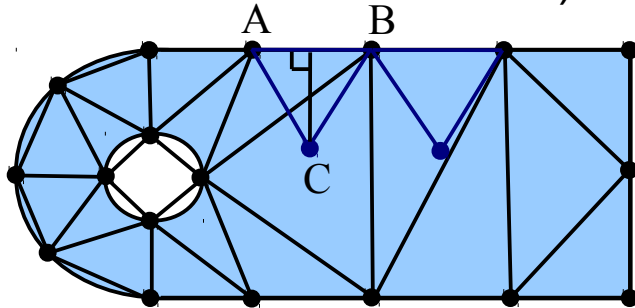


# Géométrie Algorithmique

## Triangulation de Delaunay

- Par avance de front (D.L. Marcum)

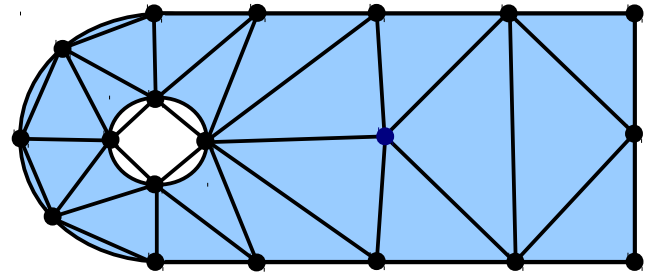
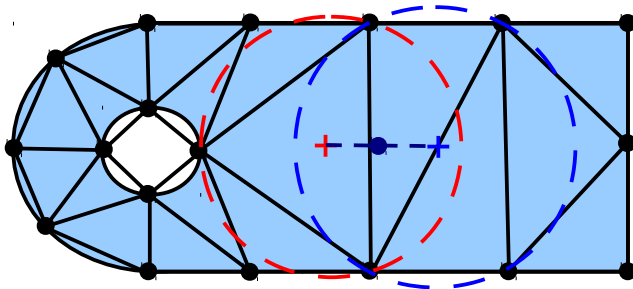
On insère des sommets depuis le bord en maintenant une structure pour le front. La position des sommets est idéale (jusqu'à ce que les fronts se rencontrent)



# Géométrie Algorithmique

## Triangulation de Delaunay

- Milieu du segment du diagramme de Voronoï reliant les centre des cercles circonscrit de deux triangles voisins. (S. Rebay, 1993)

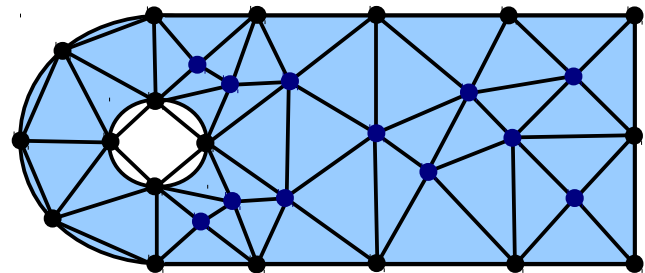
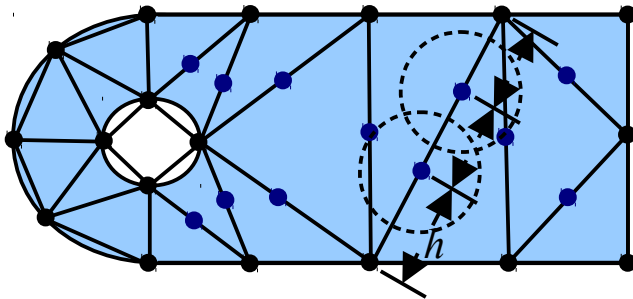


S. Rebay. *Efficient unstructured mesh generation by means of Delaunay triangulation and Bowyer-Watson algorithm*. Journal of Computational Physics, 106:25–138, 1993.

# Géométrie Algorithmique

## Triangulation de Delaunay

- Le long d'arêtes existantes - il faut vérifier que des sommets sur des arêtes voisines ne sont pas trop proches (P.L. George, 1991)



# Géométrie Algorithmique

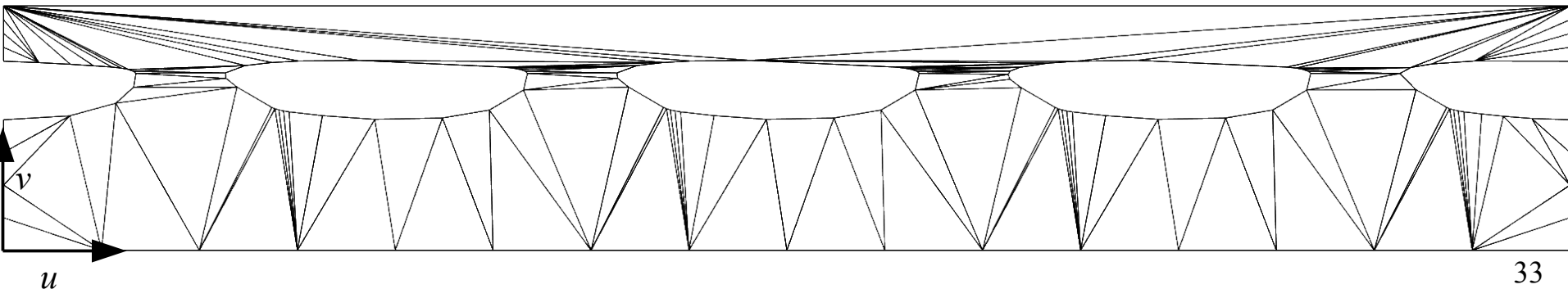
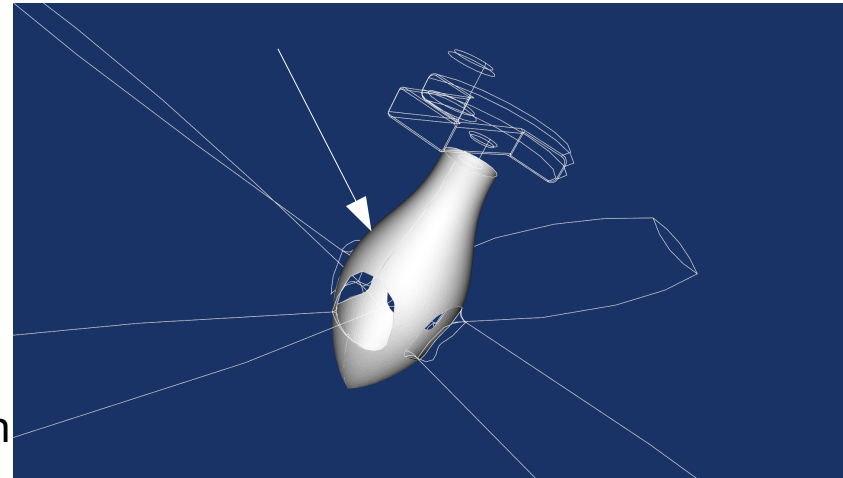
## Triangulation de Delaunay

### ■ Projet 1

Triangler (mailler) un domaine correspondant à une surface 3D (flèche), avec une taille imposée constante.

Particularités :

- Calculs dans un espace paramétrique (la notion de distance n'est pas triviale !)
- Réflexion sur la notion de triangulation de Delaunay dans cet espace
- Utiliser une méthode d'insertion de point au choix

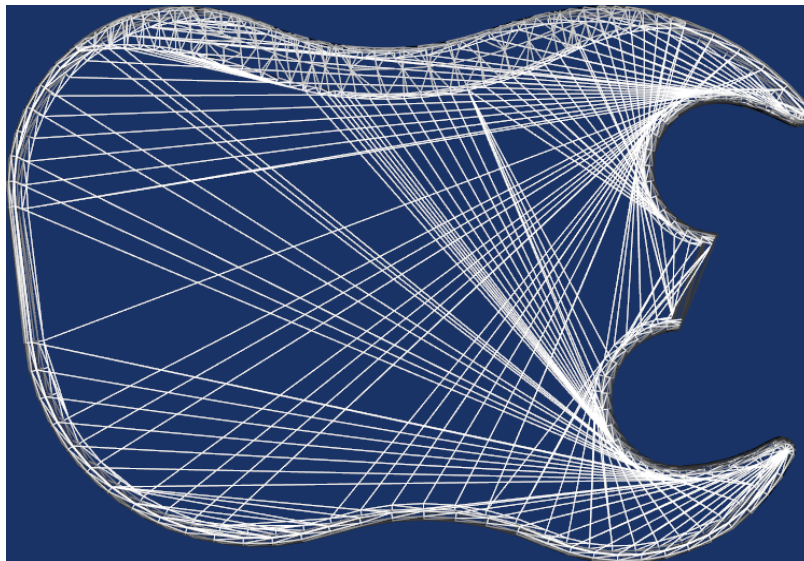




# Géométrie Algorithmique

## Triangulation de Delaunay

- **Projet 2**
  - « Amélioration » d'une triangulation surfacique
  - Triangles les plus équilatéraux
  - Triangles de taille uniforme
    - Raffinement / décimation



# Géométrie Algorithmique

## Triangulation de Delaunay

- **Projet 3**
  - Construction d'une triangulation de Delaunay ... en 3D.
  - Donnée : ensemble de points en 3D formant un volume délimité par des triangles
  - Bonus – récupérer la frontière !

# Géométrie Algorithmique

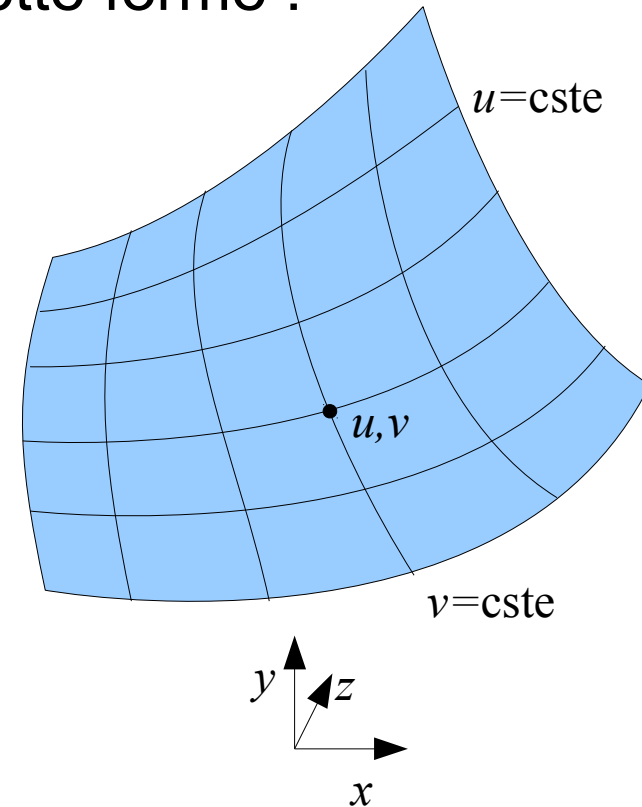
## Triangulation de Delaunay

- Une surface se représente sous cette forme :

$$\vec{P}(u, v) = \begin{cases} x = f(u, v) \\ y = g(u, v) \\ z = h(u, v) \end{cases}$$

$u, v$  sont deux paramètres réels

- On obtient tous les points de la surface en faisant varier  $u$  et  $v$ .
- La paramétrisation n'est pas unique !
- Les paramètres peuvent être bornés (limite la surface)



# Géométrie Algorithmique

## Triangulation de Delaunay

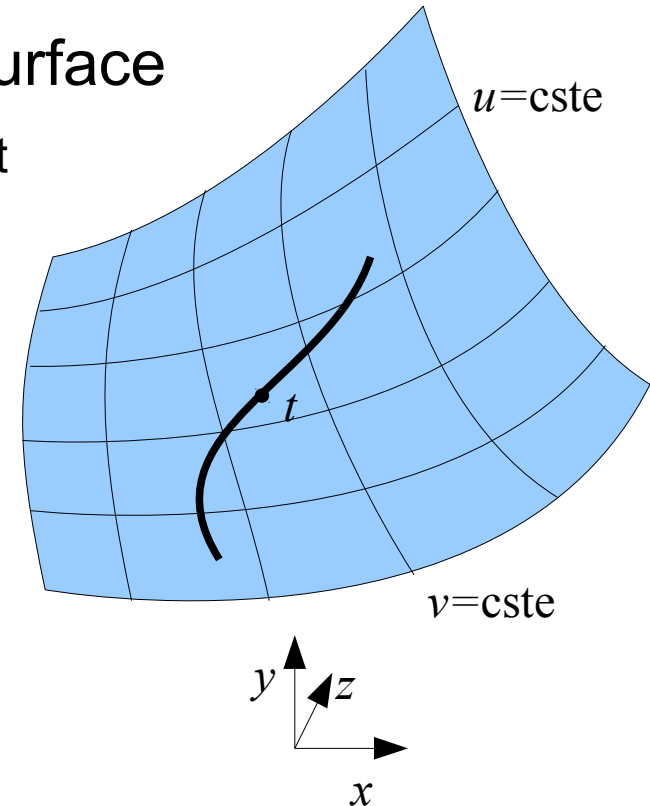
- On peut définir une courbe sur la surface

Espace paramétrique

Espace ambiant

$$\vec{\Gamma}^{uv}(t) : \begin{cases} u = u(t) \\ v = v(t) \end{cases} \longrightarrow \vec{P}(u, v) : \begin{cases} x = f(u, v) \\ y = g(u, v) \\ z = h(u, v) \end{cases}$$

$$\vec{\Gamma}(t) : \begin{cases} x = f(u(t), v(t)) \\ y = g(u(t), v(t)) \\ z = h(u(t), v(t)) \end{cases}$$



# Géométrie Algorithmique

## Triangulation de Delaunay

- Géométrie différentielle pour les surfaces paramétriques

- Position  $P$  :  $\vec{P}(u, v) = \begin{bmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{bmatrix} \quad \vec{P}^u = \frac{\partial \vec{P}}{\partial u} \quad \vec{P}^{uv} = \frac{\partial^2 \vec{P}}{\partial u \partial v} \quad \dots$

- Vecteurs tangents unitaires  $T^u$  et  $T^v$  :

$$\vec{T}^u(u, v) = \frac{\partial \vec{P}}{\partial u} \cdot \left| \frac{\partial \vec{P}}{\partial u} \right|^{-1} = \frac{\vec{P}^u}{|\vec{P}^u|} \quad \vec{T}^v(u, v) = \frac{\partial \vec{P}}{\partial v} \cdot \left| \frac{\partial \vec{P}}{\partial v} \right|^{-1} = \frac{\vec{P}^v}{|\vec{P}^v|}$$

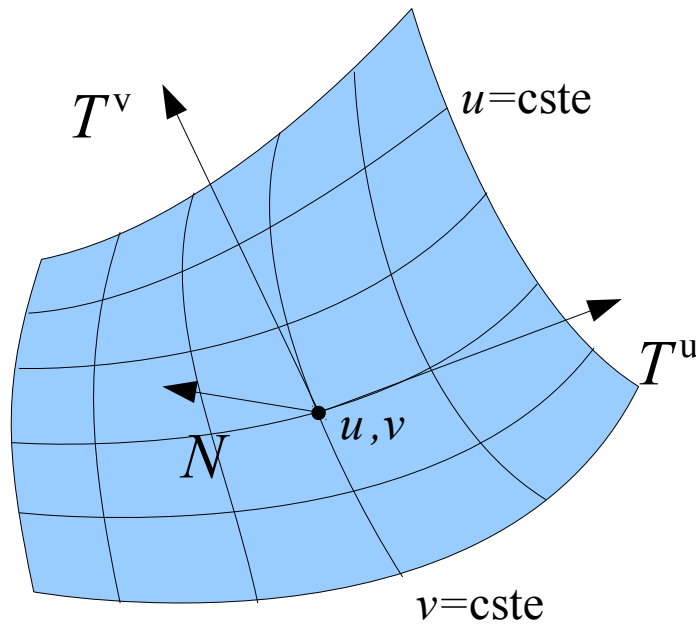
- Ces vecteurs ne sont pas perpendiculaires !

# Géométrie Algorithmique

## Triangulation de Delaunay

■

$$N(u, v) = \frac{Norm(u, v)}{|Norm(u, v)|} \text{ avec } Norm(u, v) = T^u \times T^v \text{ ou } P^u \times P^v$$



# Géométrie Algorithmique

## Triangulation de Delaunay

- Aire d'une surface

$$A = \iint_S dS$$

$$dS = |du \cdot P^u \times dv \cdot P^v| = |P^u \times P^v| du dv$$

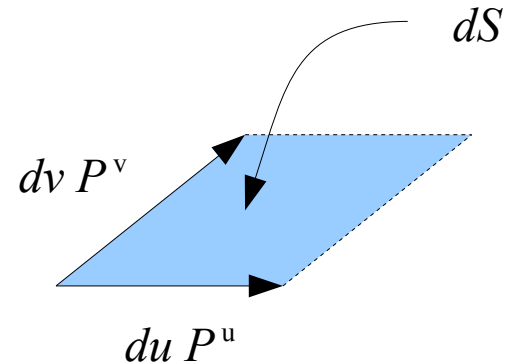
- 1<sup>ère</sup> forme fondamentale

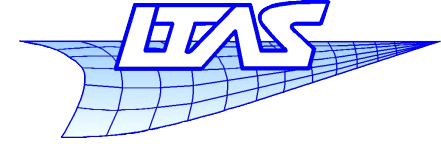
Autre notation de l'aire d'une surface

$$|\vec{a} \times \vec{b}|^2 = (\vec{a} \cdot \vec{a}) \cdot (\vec{b} \cdot \vec{b}) - (\vec{a} \cdot \vec{b})^2 \quad \leftarrow \boxed{\text{Identité de Lagrange}}$$

$$dS = \sqrt{(eg - f^2)} du dv \quad \text{avec } e = P^u \cdot P^u, \quad f = P^u \cdot P^v, \quad g = P^v \cdot P^v$$

$$A = \iint_D \sqrt{(eg - f^2)} du dv$$





## Triangulation de Delaunay

- Calcul de la longueur d'une courbe sur une surface

$$\vec{P}(u, v): \begin{cases} x = f(u, v) \\ y = g(u, v) \\ z = h(u, v) \end{cases} \quad \begin{aligned} P^u &= \frac{\partial P(u, v)}{\partial u} \\ P^v &= \frac{\partial P(u, v)}{\partial v} \end{aligned}$$

$$\vec{\Gamma}^{uv}(t): \begin{cases} u = u(t) \\ v = v(t) \end{cases} \quad \vec{\Gamma}(t): \begin{cases} x = f(u(t), v(t)) \\ y = g(u(t), v(t)) \\ z = h(u(t), v(t)) \end{cases} \quad \begin{aligned} \Gamma' &= \frac{dP(u(t), v(t))}{dt} \\ u' &= \frac{du(t)}{dt} \quad \dots \end{aligned}$$

$$d\Gamma^{uv}(t) = \begin{pmatrix} du \\ dv \end{pmatrix} = \begin{pmatrix} u' dt \\ v' dt \end{pmatrix}$$



# Géométrie Algorithmique

## Triangulation de Delaunay

- Calcul de la longueur d'une courbe sur une surface

$$L = \int_a^b \left| \vec{\Gamma}'(t) \right| dt = \int_a^b \sqrt{\quad}$$

avec  $e = P^u \cdot P^u$  ,  $f = P^u \cdot P^v$  ,  $g = P^v \cdot P^v$

# Géométrie Algorithmique

## Triangulation de Delaunay

- Si on pose

$$ds = \sqrt{e u'(t)^2 + 2 f u'(t) v'(t) + g v'(t)^2} dt$$

ce qui revient à :

( on a  $L = \int_{s(a)}^{s(b)} ds = s(b) - s(a)$  ), on a en fait une forme quadratique :

$$E u'(t)^2 + 2 F u'(t) v'(t) + G v'(t)^2 = \begin{pmatrix} u'(t) & v'(t) \end{pmatrix} \begin{pmatrix} E & F \\ F & G \end{pmatrix} \begin{pmatrix} u'(t) \\ v'(t) \end{pmatrix}$$

$$L = \int_a^b \sqrt{\begin{pmatrix} u'(t) & v'(t) \end{pmatrix} \begin{pmatrix} e & f \\ f & g \end{pmatrix} \begin{pmatrix} u'(t) \\ v'(t) \end{pmatrix}} dt$$

# Géométrie Algorithmique

## Triangulation de Delaunay

- La 1<sup>ère</sup> forme fondamentale est l'application

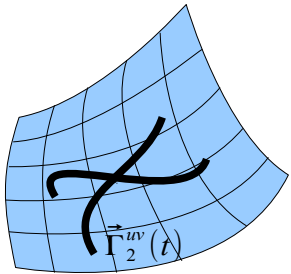
$$\vec{\Gamma}_1^{uv}(t) : \begin{cases} u = u_1(t) \\ v = v_1(t) \end{cases}$$

$$\vec{\Gamma}_2^{uv}(t) : \begin{cases} u = u_2(t) \\ v = v_2(t) \end{cases}$$

$$\phi_1(d\Gamma_1^{uv}, d\Gamma_2^{uv}) = \begin{pmatrix} du_1 & dv_1 \end{pmatrix} \begin{pmatrix} e & f \\ f & g \end{pmatrix} \begin{pmatrix} du_2 \\ dv_2 \end{pmatrix} = \begin{pmatrix} du_1 & dv_1 \end{pmatrix} M_1 \begin{pmatrix} du_2 \\ dv_2 \end{pmatrix}$$

avec  $e = P^u \cdot P^u$ ,  $f = P^u \cdot P^v$ ,  $g = P^v \cdot P^v$ ,  $du_1 = \frac{\partial u_1(t)}{\partial t}$ ,  $dv_1 = \frac{\partial v_1(t)}{\partial t}$  ...

- C'est une forme bilinéaire symétrique qui permet de « mesurer » des distances réelles à partir de variations dans l'espace paramétrique...
- La matrice  $M_1$  est la représentation du tenseur métrique.



$$L = \int_a^b \sqrt{\phi_1(d\Gamma^{uv}, d\Gamma^{uv})} dt$$

$$A = \iint_D \sqrt{\det M_1} du dv$$

- Angle « réel » entre deux courbes sécantes sur une surface paramétrique :

$$\cos \alpha = \frac{\phi_1(d\Gamma_1^{uv}, d\Gamma_2^{uv})}{\sqrt{\phi_1(d\Gamma_1^{uv}, d\Gamma_1^{uv}) \phi_1(d\Gamma_2^{uv}, d\Gamma_2^{uv})}}$$

# Géométrie Algorithmique

## Triangulation de Delaunay

- Revenons à nos moutons !!!

- Dans notre cas, on va mesurer les distances le long de segments de droites (les arêtes).

Ces segments sont définis dans l'espace paramétrique  $(u,v)$ , et munis d'une paramétrisation (e.g.  $t$  compris entre 0 et 1)

$$a^s(t) = (1-t) \cdot p_0^s + t \cdot p_1^s$$

Leur longueur se calcule avec la formule suivante :

$$L = \int_a^b \sqrt{\begin{pmatrix} u'(t) & v'(t) \end{pmatrix} \begin{pmatrix} e & f \\ f & g \end{pmatrix} \begin{pmatrix} u'(t) \\ v'(t) \end{pmatrix}} dt$$

$$L(a) = \int_0^1 \sqrt{\begin{pmatrix} p_1^s - p_0^s \end{pmatrix}^T \begin{pmatrix} e & f \\ f & g \end{pmatrix} \begin{pmatrix} p_1^s - p_0^s \end{pmatrix}} dt$$

- Si le tenseur métrique est constant, on peut faire disparaître

l'intégrale et ainsi simplifier : 
$$L(a) = \sqrt{\begin{pmatrix} p_1^s - p_0^s \end{pmatrix}^T \begin{pmatrix} e & f \\ f & g \end{pmatrix} \begin{pmatrix} p_1^s - p_0^s \end{pmatrix}}$$

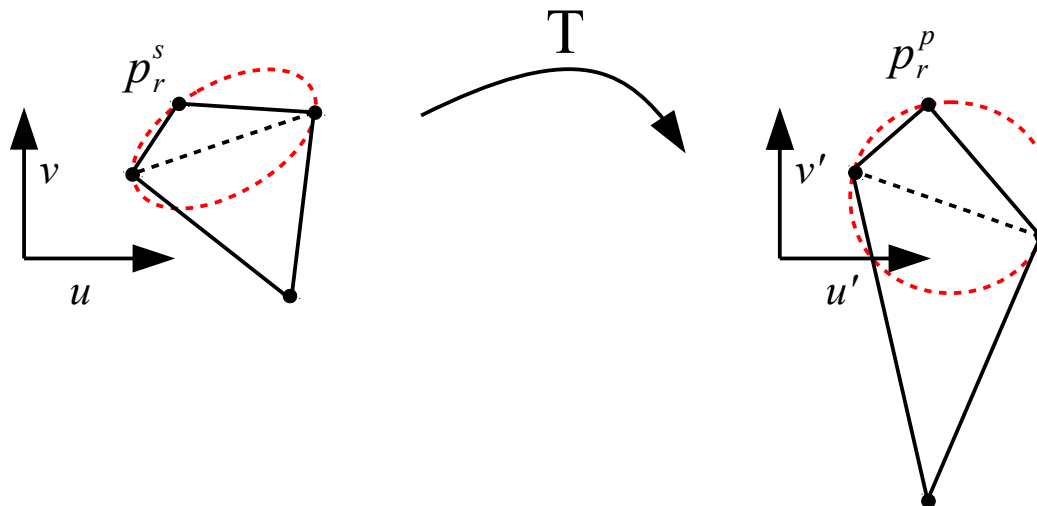
# Géométrie Algorithmique

## Triangulation de Delaunay

- Test de la légalité d'une arête

Revient à déterminer la courbe située à une distance constante d'un point ( le « centre ») et passant par trois points (dont le point inséré)

- Dans le cas général, c'est extrêmement coûteux, car le tenseur métrique est variable.
- Si celui ci est constant, alors la courbe est une ellipse...
- On fait un changement de coordonnées  $T$  (consiste en une rotation et une mise à l'échelle) pour ramener le tout dans un plan où les mesures sont euclidiennes, en utilisant le tenseur métrique...



# Géométrie Algorithmique

## Triangulation de Delaunay

- Définition de la transformation T

$$L(a) = \sqrt{(p_1^s - p_0^s)^T \begin{pmatrix} e & f \\ f & g \end{pmatrix} (p_1^s - p_0^s)} = \sqrt{(p_1^s - p_0^s)^T M_1 (p_1^s - p_0^s)}$$

On désire que les mesures se fassent dans un espace euclidien à déterminer. Dans cet espace, les coordonnées sont  $p_0^p$  et  $p_1^p$  et le tenseur métrique est l'identité I.

$$L(a) = \sqrt{(p_1^s - p_0^s)^T M_1 (p_1^s - p_0^s)} = \sqrt{(p_1^p - p_0^p)^T I (p_1^p - p_0^p)}$$

Posons  $J = \begin{pmatrix} e & f \\ f & g \end{pmatrix}$ , alors il vient

$$\begin{aligned} L(a) &= \sqrt{(p_1^s - p_0^s)^T J^T \cdot J (p_1^s - p_0^s)} = \sqrt{(p_1^s - p_0^s)^T J^T \cdot I \cdot J (p_1^s - p_0^s)} \\ &= \sqrt{(J \cdot p_1^s - J \cdot p_0^s)^T I (J \cdot p_1^s - J \cdot p_0^s)} = \sqrt{(p_1^p - p_0^p)^T I (p_1^p - p_0^p)} \end{aligned}$$

On a donc  $p_0^p = J p_0^s$  et  $p_1^p = J p_1^s$ . Ici, J est la matrice de transformation de T, ou **matrice jacobienne**.

# Géométrie Algorithmique

## Triangulation de Delaunay

- Calcul de J.  
J dépend de 4 scalaires indépendants, or,  $M_1$  est symétrique et dispose de seulement 3 degrés de libertés. Le système  $M_1 = J^T \cdot J$  est donc sous déterminé. On doit trouver **une** valeur de J satisfaisante.
- Il existe une décomposition naturelle d'une matrice M en un produit de deux matrices transposées, c'est la décomposition de Cholesky qui existe toujours si la matrice M est définie positive.  
 $M = S^T \cdot S$  avec S =matrice triangulaire supérieure.

$$M_1 = \begin{pmatrix} e & f \\ f & g \end{pmatrix} = S^T \cdot S = \begin{pmatrix} s_1 & 0 \\ s_3 & s_2 \end{pmatrix} \cdot \begin{pmatrix} s_1 & s_3 \\ 0 & s_2 \end{pmatrix} = \begin{pmatrix} s_1^2 & s_1 s_3 \\ s_1 s_3 & s_2^2 \end{pmatrix}$$

$$\text{On a donc : } S = \begin{pmatrix} \sqrt{e} & \frac{f}{\sqrt{e}} \\ 0 & \sqrt{g} \end{pmatrix} = J$$

- Pour déterminer si un cercle contient un point donné, on va donc utiliser cette matrice J pour transformer les coordonnées de tous les points (4) . Il est judicieux de ramener toutes les coordonnées à des variations autour d'un point , par exemple le point inséré  $p_r$  (qui est alors l'origine du repère)

# Géométrie Algorithmique

## Triangulation de Delaunay

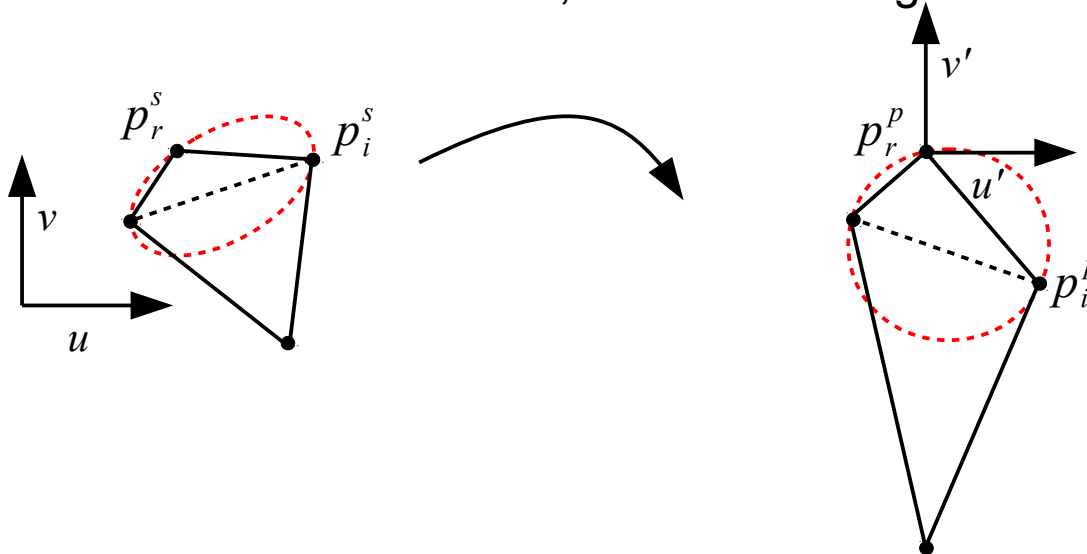
- En définitive,

$$J = \begin{pmatrix} \sqrt{e} & \frac{f}{\sqrt{e}} \\ 0 & \sqrt{g} \end{pmatrix} \text{ avec } e = P^u \cdot P^u, \quad f = P^u \cdot P^v, \quad g = P^v \cdot P^v$$

Nouvelles coordonnées :

$$p_i^p = J \cdot (p_i^s - p_r^s)$$

Dans ces coordonnées, le test sur la légalité de l'arête est classique.





# Géométrie Algorithmique

## Triangulation de Delaunay

- Définition de la surface support

Soient  $P_0 \dots P_4$  5 points de contrôle, et  $C_0 \dots C_4$  5 coefficients entiers.

$$P_0 = \begin{pmatrix} 0 \\ 15 \\ -75 \end{pmatrix}, \quad P_1 = \begin{pmatrix} 0 \\ 15 \\ -37.5 \end{pmatrix}, \quad P_2 = \begin{pmatrix} 0 \\ 45 \\ -5 \end{pmatrix}, \quad P_3 = \begin{pmatrix} 0 \\ 45 \\ 27.5 \end{pmatrix}, \quad P_4 = \begin{pmatrix} 0 \\ 0 \\ 75 \end{pmatrix}$$

$$C_0=1, \quad C_1=4, \quad C_2=6, \quad C_3=4, \quad C_4=1$$

La courbe  $C(v)$  est définie par :

$$C(v) = \sum_{i=0}^4 C_i (1-v)^i v^{(4-i)} P_i = \begin{pmatrix} C_x(v) \quad (\equiv 0) \\ C_y(v) \\ C_z(v) \end{pmatrix}$$

La surface de révolution  $S(u,v)$  est alors définie par :

$$S(u, v) = \bar{R}(u) \cdot C(v) = \begin{pmatrix} C_x(v) \cdot \cos(u) - C_y(v) \cdot \sin(u) \\ C_x(v) \cdot \sin(u) + C_y(v) \cdot \cos(u) \\ C_z(v) \end{pmatrix}$$

# Géométrie Algorithmique

## Triangulation de Delaunay

- Définition du polygone simplifié à « trianguler » dans l'espace paramétrique de la surface  $S(u,v)$

