Détection de collisions

Géométrie algorithmique - INFO-F-420

Tim Lenertz

ULB, MA1 INFO

22 avril 2014

Introduction

- Espace 2D/3D avec objets géométriques animés
- Détecter collisions de façon dynamique
- Ex. simulations physiques, jeux video, robotique, ...
- Temps réel → Algorithmes efficaces
- Calcul d'intersection, distance entre objets
- Structure de données des objets
- Subdivision de l'espace

A posteriori

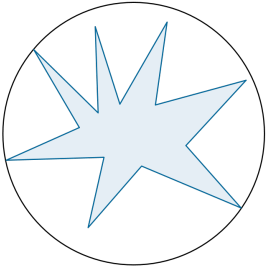
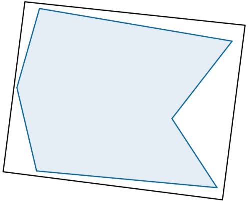
- Faire avancer simulation
- Intersection de produit
- $\rightarrow$ Corriger, traitement de collision
- ex. appliques lois cinétiques

A priori

- Prédire collisions
- Intersection ne se produisent jamais
- Calcul de temps/point d'inpact futur
- ex. controlleur système réel

Bounding boxes

- Enveloppe simple autour d'objet
- ex. rectangle, sphère, polygone convexe
- Permet exclure collision en $O(1)$

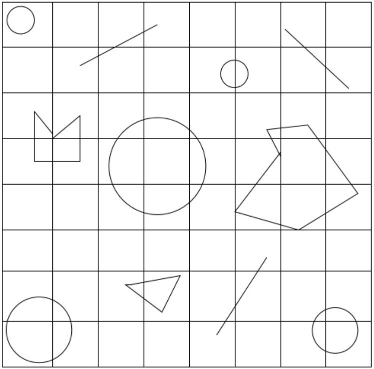



Subdivision d'espace

- Scène complexe avec bcp d'objets
- Peu de collisions possibles ($\ll n^2$)
- Regrouper objets proches / collisions potentielles
- Exclure majorité des tests

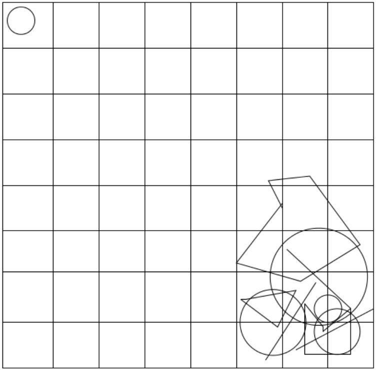
Grille uniforme

- Cellules carrées/cubiques de ême taille
- Enregistrer objets/cellule
- Tester seulement ds même cellule
- Bonne performance si objets distribués uniformement



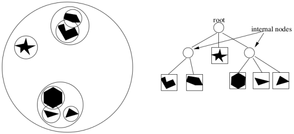
Grille uniforme (2)

- Clustering
- Cellules trop grandes/petites
- Dégénère vers $O(n^2)$



Bounding Volume Hierarchy

- Regrouper objets dans arbre (dynamique)
- Tester d'abord *bounding boxes* des groupements
- Peut atteindre $O(\log n)$



Autres optimisations

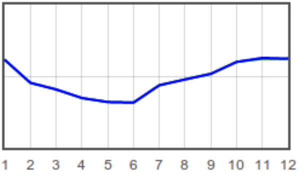
- Quadtree/Octree
- Binary Space Partition
- View Frustum Culling
- ...

Introduction

- Traité ici : polygones convexes (2D)
- Algorithmes : intersection, distance minimale
- en $O(\log n)$

Fonction unimodale

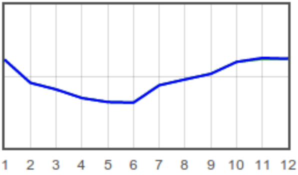
- Fonction réelle sur entiers $f : \{1, 2, ..., n\} \to \mathbb{R}$
- Existe entire $m \in [1, n]$
- f strictement $\nearrow$ ($\searrow$) en $[1, m]$ et str. $\searrow$ ($\nearrow$) en $[m + 1, n]$
- Exemple :



$$m = 6$$

Trouver maximum et minimum

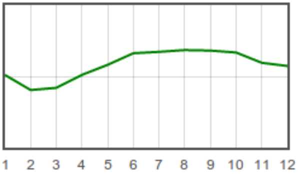
- Comparer $f(1)$ et $f(2)$, pour déterminer si $\nearrow\searrow$ ou $\searrow\nearrow$
- $\nearrow\searrow$: Minimum $= \min\{f(1), f(n)\}$, Maximum $= m$
- $\searrow\nearrow$: Maximum $= \max\{f(1), f(n)\}$, Minimum $= m$
- Recherche dichotomique : Trouver point où signe de $f(i+1) - f(i)$ change
- Complexité logarithmique $O(\log n)$



$m = 6$

Fonction bimodale

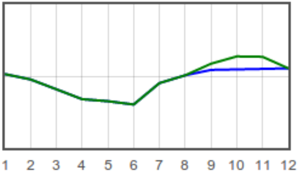
- Décalage circulaire d'une fonction unimodale
- Il existe $r \in [1, n]$ tel que $f(r), f(r+1), \ldots, f(n), f(1), f(2), \ldots, f(r-1)$ est unimodale
- Deux extréma à l'intérieur
- Soit $\nearrow\searrow\nearrow$ et $f(1) > f(n)$, ou $\searrow\nearrow\searrow$ et $f(1) < f(n)$



$\min = f(2); \max = f(8)$

Trouver maximum et minimum

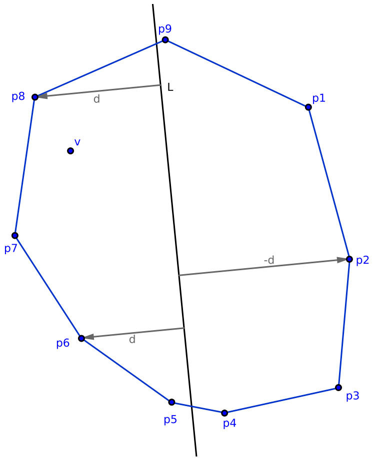
- Pour le cas $f(1) < f(n)$ et $\searrow\nearrow\searrow$:
- Si $f(2) < (1)$:
  - Soit $T(x)$ droite de $(1, f(1))$ à $(n, f(n))$ $(\nearrow)$
  - Soit $g(x) = \min\{T(x), f(x)\}$
  - g est unimodale, $\min f = \min g$
  - $\max f$ dans séq. unimodale $f(m+1), f(m+2), ..., f(n)$
  - Donc on trouve min et max en $O(\log n)$
- Si $f(2) > f(1)$: $f(1)$ est minimum, maximum est dans $f(2)...f(n)$ (unimodal)



f en vert, g en vert+bleu

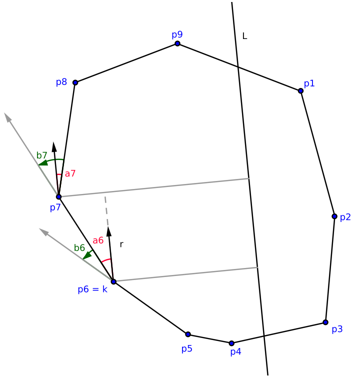
Distances orientées

- Soit P polygone convexe
- $\{p_1, p_2, \ldots, p_n\}$ points en sens horlogique
- $d(p_i, L) =$ distance orthogonale de p_i à L
- Distance orientée $h(p_i, L, v)$:
  - $= d(p_i, L)$ si p_i et v sur même côté de P
  - $= -d(p_i)$ sinon
- $h(i) = h(p_i, L, v)$ bimodale



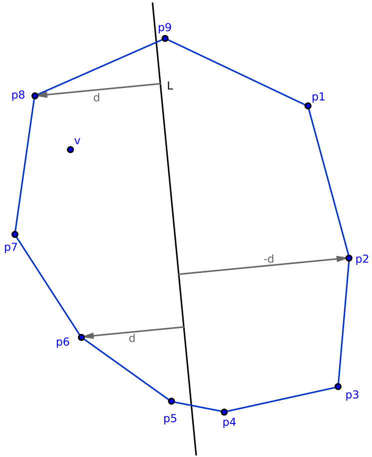
Démonstration bimodalité de h

- Soit p_k point qui minimise h
- Montrer que $f(k), f(k+1), \ldots, f(k-1)$ est unimodale (indices modulo n) :
  - r vecteur directeur de L (avec $\angle(r, p_i p_{i+1}) < \pi$)
  - $h(i+1) = h(i) + |p_i p_{i+1}| \sin a_i$
  - $a_{i+1} = a_i - b_{i+1}$
  - $b_i < \pi$ car P est convexe
  - $\sum b_i = 2\pi$
  - $\sin a_k, \sin a_{k+1}, \ldots \sin_{k-1}$ positive, puis négative
  - Donc $f(k), f(k+1), \ldots, f(k-1)$ unimodale $\square$



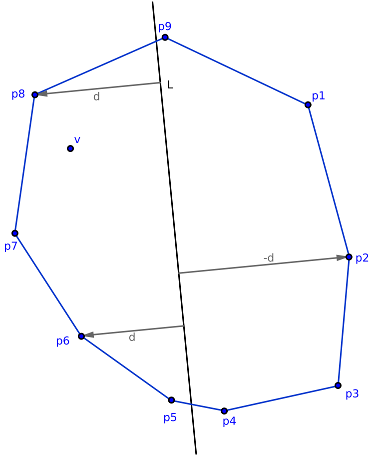
Algorithme IGL

- Trouver points $P \cap L$ en temps logarithmique
- Utiliser $h(p_i, L, p_1)$ bimodale
- Soit p_k minimum de h
- Si $h(k) > 0$:
  - Tous les points sur même côté de P
  - $(=$ côté de $p_1)$
  - donc pas d'intersection
- Si $h(k) = 0$:
  - p_k est unique point d'intersection
  - Autres points tous sur même côté



Algorithme IGL (2)

- Si $h(k) < 0$:
  - 2 points d'intersection
  - Sur segments $p_i p_{i+1}$ pour lesquels $h(i) \times h(i+1) < 0$
  - Recherche dichotomique sur séq. monotones $h(k), h(k+1), \dots, h(n)$ et $h(1), h(2), \dots, h(k)$
  - (p_1 côté opposé de p_k)
  - Calculer $p_i p_{i+1} \cap L$
- Donc points d'intersection trouvés en $O(\log n)$ $\square$



Démo

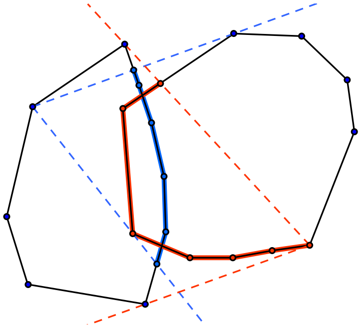
Généralisations

- Généralisation de l'algo précédent
- Sur objets convexes 2D et 3D
- Temps logarithmiques
- On intersecte l'intérieur des objets
- Complexités :

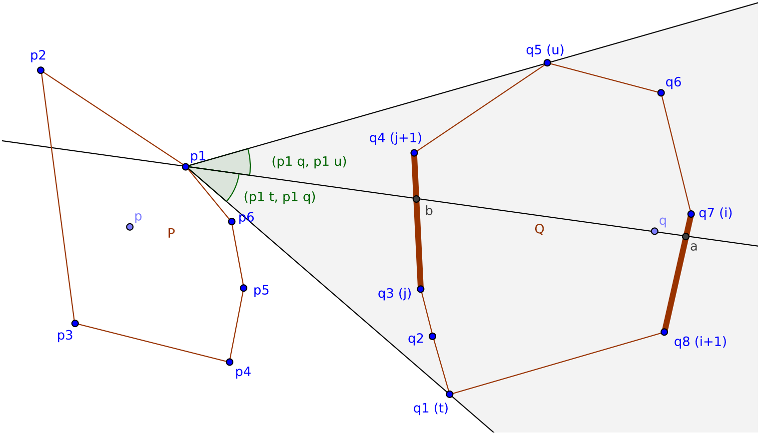
droite-droite $O(1)$
droite-polygone $O(\log n)$
droite-plan $O(1)$
droite-polyhédron $O(\log^2 n)$
polygone-polygone $O(\log n)$
polygone-plan $O(\log n)$
polygone-polyhédron $O(\log^2 n)$
plan-plan $O(1)$
plan-polyhédron $O(\log^2 n)$
polyhédron-polyhédron $O(\log^3 n)$

Algorithme IGG

- Reduire à intersection de deux polylignes
- Elimination binaire
- $O(\log(n+m))$

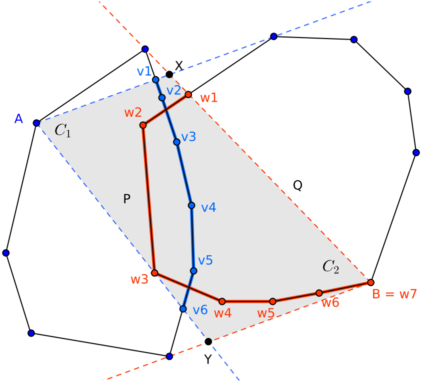


Phase initiale (1)



- $q \in Q$ quelconque
- $a, b = p_1 q \cap Q$ (algorithme IGL)
- Si $p_1 \in ab$, intersection en p_1

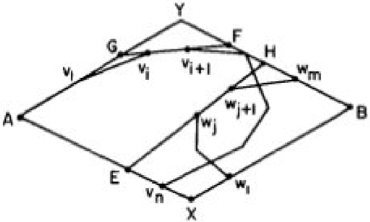
Phase initiale (2)



- Déduire polylignes L_v et L_w
- Intersections $C_1 \cap Q$ + points de Q (même pour P)
- Possible en $O(\log n)$ par IGL
- P et Q intersectent ssi L_v et L_w intersectent

Démo

Phase itérative

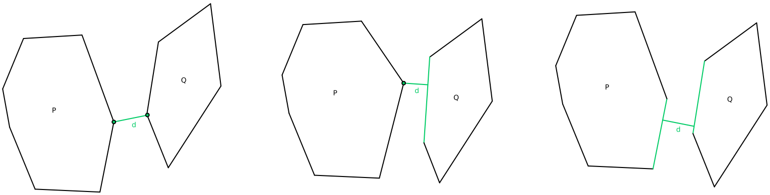


- Soit $n = |L_v|$, $m = |L_w|$
- Tant que $n, m > 5$ (sinon, différent algorithme)
- $i = \lfloor \frac{n}{2} \rfloor$ et $j = \lfloor \frac{m}{2} \rfloor$
- $F, G = v_i v_{i+1} \cap AYBX$ (avec $v_{i+1} \in v_i F$)
- $E, H = w_j v_{j+1} \cap AYBX$ (avec $w_{j+1} \in v_j H$)
- Eliminer à chaque étape moitié de L_v et/ou L_w

Détail algorithme

Distance minimale de deux polygones convexes

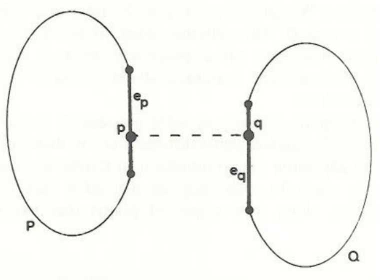
- Segment d de longueur minimale qui relie P et Q
- Détection collisions *à priori*
- Algorithme en $O(\log n + \log m)$
- 3 cas possibles : point-point, point-droite, droite-droite
- droite-droite : parallèles, $=$ point-droite



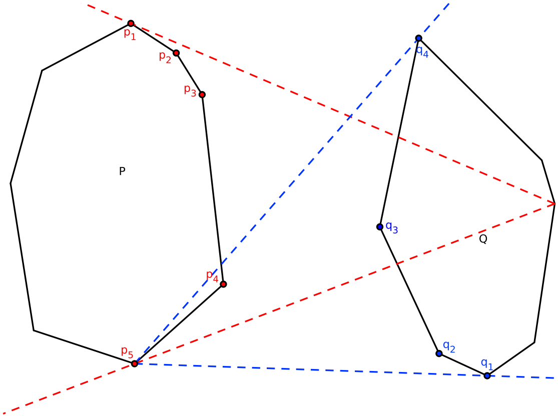
Preuve 3 cas possibles

- Soit $d = pq$
- p, q doivent être sur bord des polygones
- p et/ou q doit être coin de polygone
- Eliminer cas où p **et** q sont sur segments :
  - Soit $p \in e_p$ et $q \in a_q$
  - Alors $e_p \parallel e_q$
  - $\exists$ projection orthogonale de point de e_p (e_q) sur e_q (e_p)

$\square$

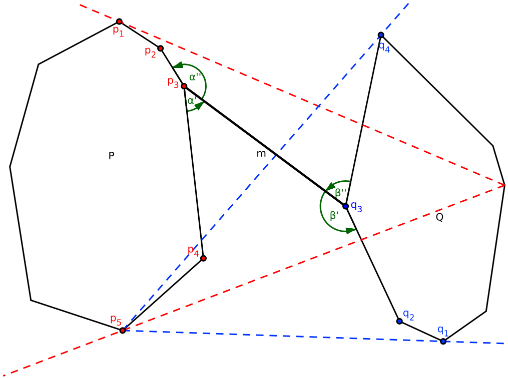


Phase initiale



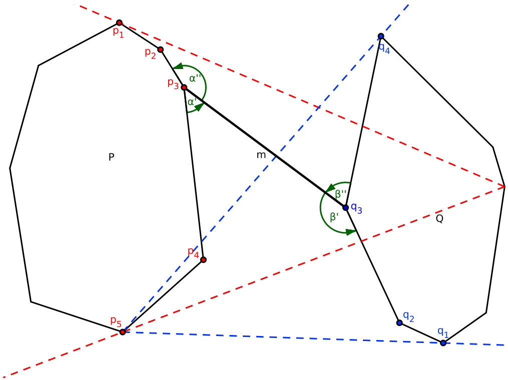
- Déterminer C_1 et C_2
- Polylignes L_p et L_q dans P et Q
- d doit relier L_p et L_q

Phase itérative



- $i = \lfloor \frac{n}{2} \rfloor$ et $j = \lfloor \frac{m}{2} \rfloor$, $m = p_i q_j$
- $\alpha' + \alpha'' \geq \pi$ et donc $\alpha' \geq \frac{\pi}{2}$ ou $\alpha'' \geq \frac{\pi}{2}$
- $\alpha' + \beta' \leq \pi$ implique $\alpha' < \beta''$
- $\alpha' + \beta' > \pi$ ou $\alpha'' + \beta'' > \pi$

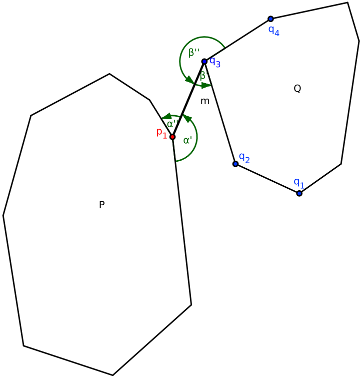
Phase itérative (2)



- Eliminer moitié de L_p et/ou L_q
- Jusqu'à $|L_p|, |L_p| \leq 2$
- Distinguer cas selon $|L_p|$ et $|L_p|$

Cas 1 : $|L_p| = 1$

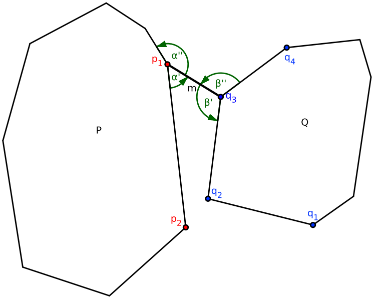
- Si $\beta' \geq \frac{\pi}{2}$: $q_{\text{first}} \leftarrow q_j$
- Si $\beta'' \geq \frac{\pi}{2}$: $q_{\text{last}} \leftarrow q_j$
- $|L_q| = 1$ est symétrique
- Au moins une condition doit être vraie
- $\leftarrow$ On élimine tjs moitié



Cas 2 : $|L_p| = 2$

Si m sort de P :

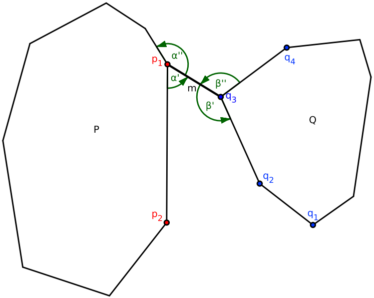
1. Si $\alpha' + \beta' > \pi$:
   Si $\alpha' \geq \frac{\pi}{2}$, $p_{\text{first}} \leftarrow p_2$
   Si $\beta' \geq \frac{\pi}{2}$, $q_{\text{first}} \leftarrow q_j$
2. Si $\beta'' \geq \frac{\pi}{2}$: $q_{\text{last}} \leftarrow q_j$
3. ...



Cas 2 : $|L_p| = 2$

Si m sort de P :

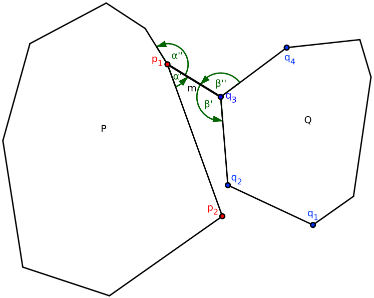
1. Si $\alpha' + \beta' > \pi$:
   Si $\alpha' \geq \frac{\pi}{2}$, $p_{\text{first}} \leftarrow p_2$
   Si $\beta' \geq \frac{\pi}{2}$, $q_{\text{first}} \leftarrow q_j$
2. Si $\beta'' \geq \frac{\pi}{2}$: $q_{\text{last}} \leftarrow q_j$
3. ...



Cas 2 : $|L_p| = 2$

Si m sort de P :

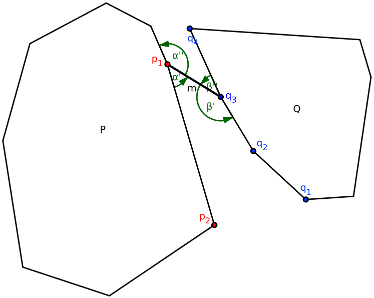
1. Si $\alpha' + \beta' > \pi$:
   Si $\alpha' \geq \frac{\pi}{2}$, $p_{\text{first}} \leftarrow p_2$
   Si $\beta' \geq \frac{\pi}{2}$, $q_{\text{first}} \leftarrow q_j$
2. Si $\beta'' \geq \frac{\pi}{2}$: $q_{\text{last}} \leftarrow q_j$
3. ...



Cas 2 : $|L_p| = 2$

Si m sort de P :

1. Si $\alpha' + \beta' > \pi$:
   Si $\alpha' \geq \frac{\pi}{2}$, $p_{\text{first}} \leftarrow p_2$
   Si $\beta' \geq \frac{\pi}{2}$, $q_{\text{first}} \leftarrow q_j$
2. Si $\beta'' \geq \frac{\pi}{2}$: $q_{\text{last}} \leftarrow q_j$
3. ...



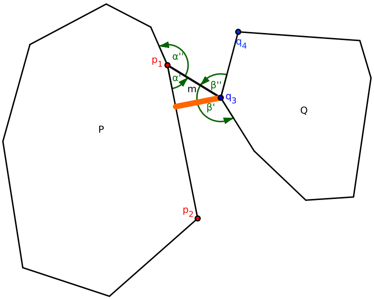
Cas 2 : $|L_p| = 2$ (2)

Si m sort de P :

1. Si $\alpha' + \beta' > \pi$:
   Si $\alpha' \geq \frac{\pi}{2}$, $p_{\text{first}} \leftarrow p_2$
   Si $\beta' \geq \frac{\pi}{2}$, $q_{\text{first}} \leftarrow q_j$
2. Si $\beta'' \geq \frac{\pi}{2}$: $q_{\text{last}} \leftarrow q_j$

   $\vdots$

3. Si $\alpha' < \beta'' < \frac{\pi}{2}$:
   - Si proj orth q_j sur $p_1 p_2$ existe : $q_{\text{last}} \leftarrow q_j$
   - Sinon : $p_{\text{last}} \leftarrow p_1$



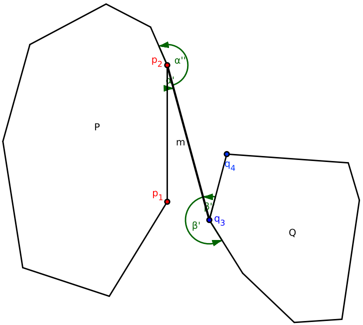
Cas 2 : $|L_p| = 2$ (2)

Si m sort de P :

1. Si $\alpha' + \beta' > \pi$:
   Si $\alpha' \geq \frac{\pi}{2}$, $p_{\text{first}} \leftarrow p_2$
   Si $\beta' \geq \frac{\pi}{2}$, $q_{\text{first}} \leftarrow q_j$
2. Si $\beta'' \geq \frac{\pi}{2}$: $q_{\text{last}} \leftarrow q_j$

   ⋮

3. Si $\alpha' < \beta'' < \frac{\pi}{2}$:
   - Si proj orth q_j sur $p_1 p_2$ existe : $q_{\text{last}} \leftarrow q_j$
   - Sinon : $p_{\text{last}} \leftarrow p_1$



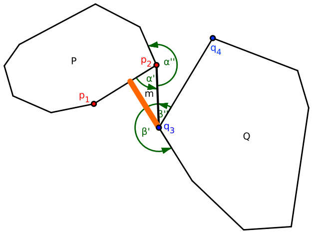
Cas 2 : $|L_p| = 2$ (2)

Si m sort de P :

1. Si $\alpha' + \beta' > \pi$:
   Si $\alpha' \geq \frac{\pi}{2}$, $p_{\text{first}} \leftarrow p_2$
   Si $\beta' \geq \frac{\pi}{2}$, $q_{\text{first}} \leftarrow q_j$
2. Si $\beta'' \geq \frac{\pi}{2}$: $q_{\text{last}} \leftarrow q_j$

   $\vdots$

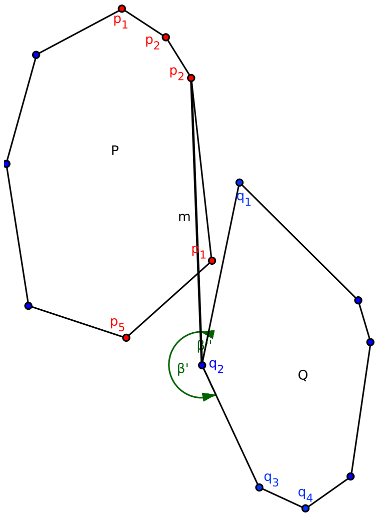
3. Si $\alpha' < \beta'' < \frac{\pi}{2}$:
   - Si proj orth q_j sur $p_1 p_2$ existe : $q_{\text{last}} \leftarrow q_j$
   - Sinon : $p_{\text{last}} \leftarrow p_1$



Cas 2 : $|L_p| = 2$ (3)

Si m entre dans P :

- $p_{\text{last}} \leftarrow p_1$
- Si $\beta' \geq \pi$: $q_{\text{first}} \leftarrow q_j$
- Si $\beta'' \geq \pi$: $q_{\text{last}} \leftarrow q_j$



Cas 3 : $|L_p| \geq 3$ et $|L_q| \geq 3$

Si m sort de P et de Q :

1. Si $\alpha' + \beta' > \pi$:
   - Si $\alpha' \geq \frac{\pi}{2}$: $p_{\text{first}} \leftarrow p_i$
   - Si $\beta' \geq \frac{\pi}{2}$: $q_{\text{first}} \leftarrow q_j$
2. Si $\alpha'' + \beta'' > \pi$:
   - Si $\alpha'' \geq \frac{\pi}{2}$: $p_{\text{last}} \leftarrow p_i$
   - Si $\beta'' \geq \frac{\pi}{2}$: $q_{\text{last}} \leftarrow q_j$

Si m entre dans P ou Q :
similaire au cas précédent