CHAPITRE VIII

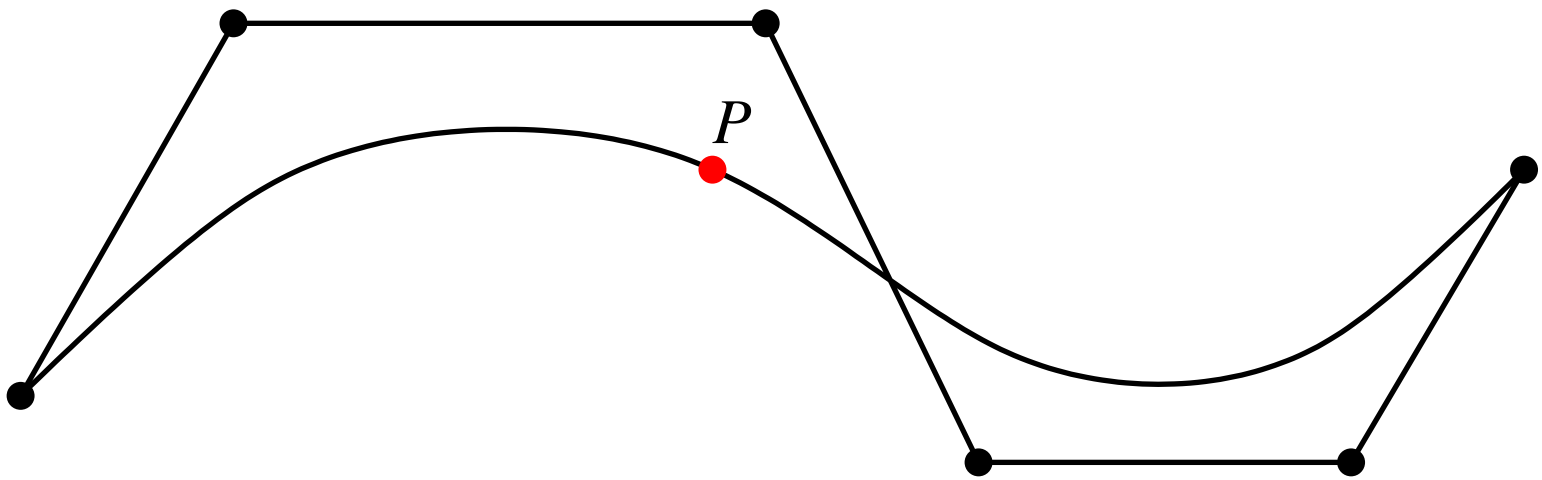
Algorithmes avancés

Sommaire

- Projection et inversion sur une courbe,
- Projection et inversion sur une surface,
- Intersection surface-surface

Projection et inversion sur une courbe

Soit une courbe NURBS, et soit un point P sur la courbe, on veut déterminer le paramètre u tel que $P = C(u)$



c'est le problème de l'*inversion*.

Projection et inversion sur une courbe

Première constatation, pour $p \leq 4$, ce problème peut être résolu exactement, de la façon suivante :

1. Décomposer la courbe en bouts de Bézier.
2. On sélectionne les intervalles qui contiennent potentiellement le point par la propriété de l'enveloppe convexe.
3. Chaque segment peut être convertie en base de puissances.

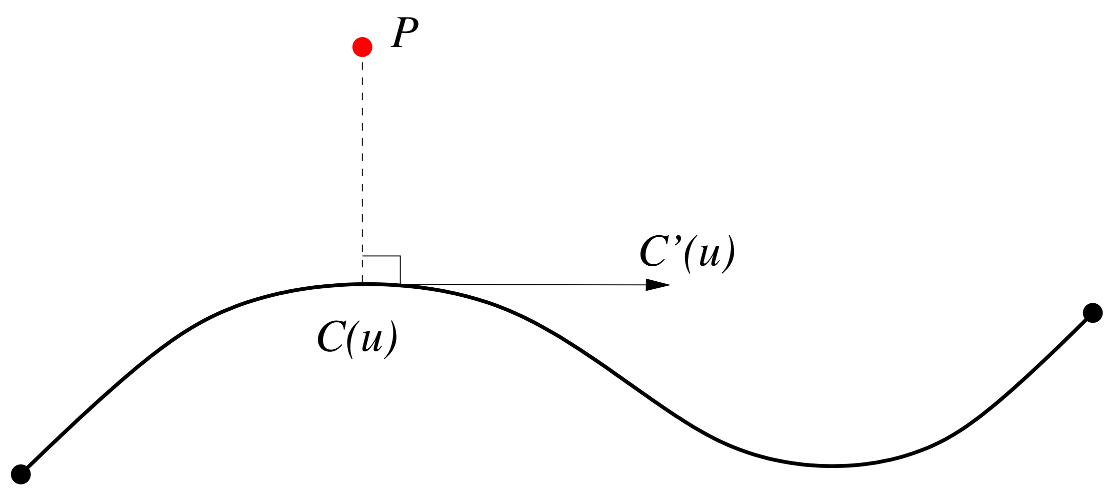
$$\sum_{i=j+p+1}^{j} N_{i,p} P_i \longrightarrow \sum a_i u^i \quad \text{avec } j \text{ l'intervalle}$$

Projection et inversion sur une courbe

4. On trouve les zéros du polynôme pour (x, y, z)
   $\longrightarrow$ On obtient 3 équations qui doivent être satisfaites simultanément et si c'est le cas à une tolérance donnée, le u qui satisfait les 3 équations est le paramètre cherché.

Projection et inversion sur une courbe

L'autre approche, qui est purement numérique, consiste à faire des *itérations de Newton* pour déterminer le zéro du système.



$$C'(u) \cdot (P - C(u)) = 0 = f(u)$$

$f(u)$ est un système non-linéaire pour lequel on cherche un zéro.

Projection et inversion sur une courbe

$$u_{i+1} = u_i - \frac{f(u_i)}{f'(u_i)}$$

$$u_{i+1} = u_i - \frac{C'(u_i) \cdot (C(u_i) - P)}{C''(u_i) \cdot (C(u_i) - P) + |C'(u_i)|^2}$$

Deux problèmes :

1. Trouver un bon point de départ

2. Arrêter à temps

Projection et inversion sur une courbe

Pour trouver un bon point de départ on peut:

1. utiliser la propriété de l'enveloppe convexe pour sélectionner les intervalles candidats,
2. discrétiser chaque intervalle avec un nombre de points approprié
3. trouver le point de discrétisation le plus proche du point que l'on veut projeter, et
4. utiliser ce point de discrétisation comme point de départ de l'itération de Newton.

Projection et inversion sur une courbe

Critère d'arrêt :

La correction est suffisament petite

$$|(u_{i+1} - u_i)C'(u_i)| \leq \epsilon_1 \tag{1}$$

Pour une inversion

$$|P - C(u_i)| \leq \epsilon_1 \tag{2}$$

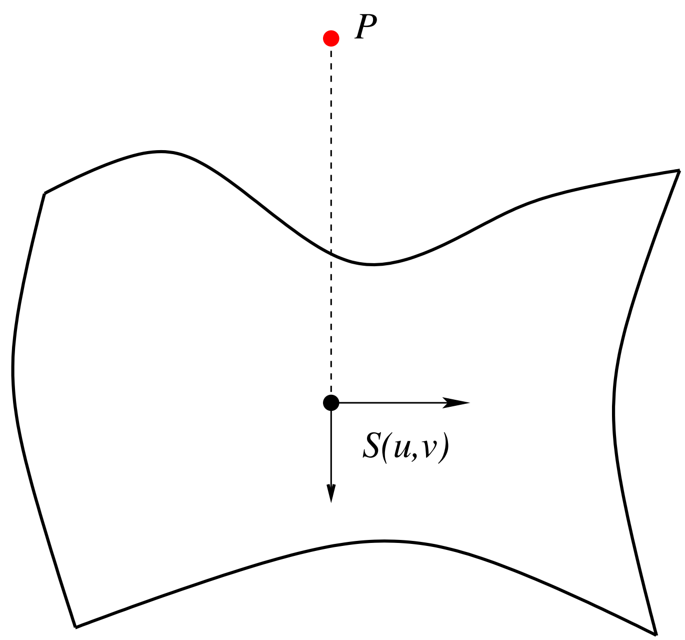
Orthogonalité de la projection

$$\frac{|C'(u_i) \cdot (C(u_i) - P)|}{|C'(u_i)||C(u_i) - P|} \leq \epsilon_2 \tag{3}$$

Pour décider d'arrêter, les critères (1) et (2) doivent être satisfaits simultanément.

Projection et inversion sur une surface

L'approche basée sur les itérations de Newton reste valable.



Projection et inversion sur une surface

$$f(u,v) = \vec{r}(u,v) \cdot S_u(u,v) = 0$$

$$g(u,v) = \vec{r}(u,v) \cdot S_v(u,v) = 0$$

où

$$\vec{r}(u,v) = S(u,v) - P$$

Projection et inversion sur une surface

$$\delta_i = \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = \begin{bmatrix} u_{i+1} - u_i \\ v_{i+1} - v_i \end{bmatrix} \text{(correction)}$$

$$J_i = \begin{bmatrix} f_u & f_v \\ g_u & g_v \end{bmatrix} \text{(dérivé)}$$

$$\kappa_i = -\begin{bmatrix} f(u_i, v_i) \\ g(u_i, v_i) \end{bmatrix}$$

Projection et inversion sur une surface

$$J_i \delta_i = \kappa_i$$

à partir de δ_i on obtient

$$\begin{aligned} u_{i+1} &= \Delta u + u_i \\ v_{i+1} &= \Delta v + v_i \end{aligned}$$

Projection et inversion sur une surface

Critère d'arrêt :

La correction est suffisament petite

$$|(u_{i+1} - u_i)S_u(u_i, v_i) + (v_{i+1} - v_i)S_v(u_i, v_i)| \leq \epsilon_1 \qquad (4)$$

Pour une inversion

$$|S(u_i, v_i) - P| < \epsilon_1 \qquad (5)$$

Petit cosinus

$$\frac{|S_u(u_i, v_i) \cdot (S(u_i, v_i) - P)|}{|S_u(u_i, v_i)||S(u_i, v_i) - P|} \leq \epsilon_2 \quad \text{idem pour la direction } v \qquad (6)$$

À chaque itération de Newton, il faut s'assurer que les paramètres restent à l'intérieur de l'intervalle de validité.

Intersection surface-surface

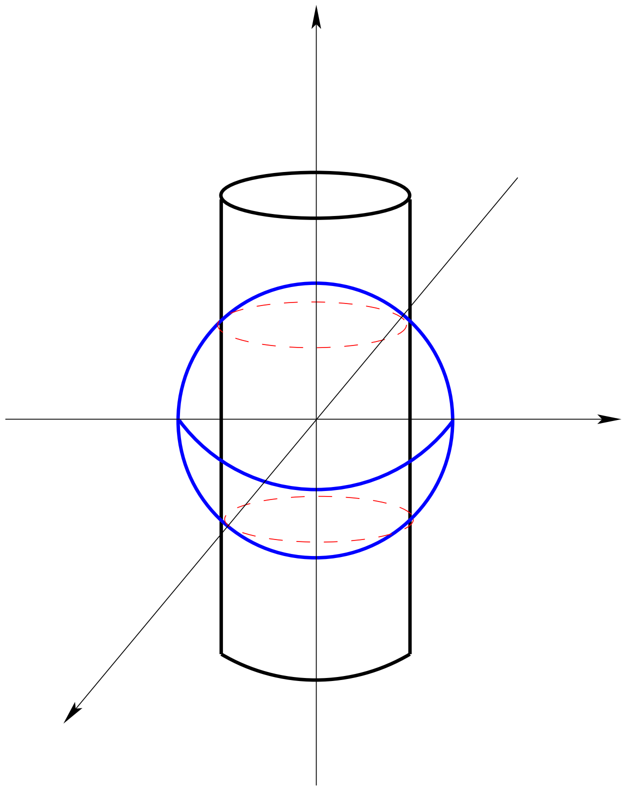
Soit 2 surfaces quelconques, on veut trouver la ou les courbes qui décrivent l'intersection entre les 2 surfaces.

Dépendant de l'application envisagée pour la courbe d'intersection, la qualité de la représentation de cette courbe peut varier d'une représentation linéaire par morceaux, à une représentation de degré élevé.

En particulier, si on travaille avec un mélange de surfaces paramétriques et implicites, ou s'il est facile de convertir l'une des surfaces paramétrique en surface implicite, le calcul exact de la courbe d'intersection devient “facile”.

Intersection surface-surface

Il s'agit alors de remplacer l'équation paramétrique dans l'équation implicite et de résoudre l'équation résultante de façon analytique.



cylindre :

$$
\begin{aligned}
X_c &= R_c \cos u \\
Y_c &= R_c \sin u \\
Z_c &= v
\end{aligned}
$$

sphère :

$$X^2 + Y^2 + Z^2 = R_s^2$$

Intersection surface-surface

$$(R_c \cos u)^2 + (R_c \sin u)^2 + v^2 = R_s^2$$

$$\begin{aligned} R_c^2 + v^2 &= R_s^2 \\ v^2 &= R_s^2 - R_c^2 \\ v &= \pm\sqrt{R_s^2 - R_c^2} \end{aligned}$$

Intersection surface-surface paramétriques

Le problème se pose principalement au niveau de l'intersection entre 2 surfaces paramétriques. Dans ce cas là, *Timmer (1977)* a proposé une approche en 3 étapes pour le calcul de l'intersection.

I - Identifier les points de départ des courbes d'intersection.

II - Poursuivre les courbes d'intersection simultanément sur les 2 surfaces.

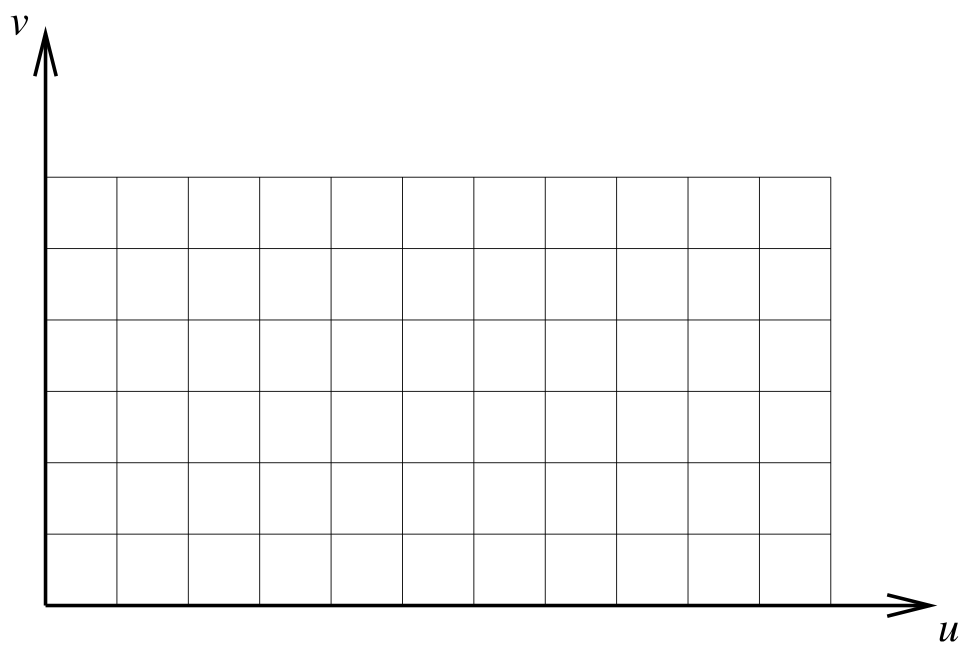
III - Ordonner les segments de courbes d'intersection pour former des boucles.

Par la suite, on peut ajouter une étape supplémentaire:

IV - Spliner les courbes paramétriques pour retrouver des courbes continues

I - Identifier les points de départ

On va choisir une des 2 surfaces et sur cette surface, on construit une grille de lignes isoparamétriques en u et en v



$S_1(u, v) \bigcap S_2(s, t)$

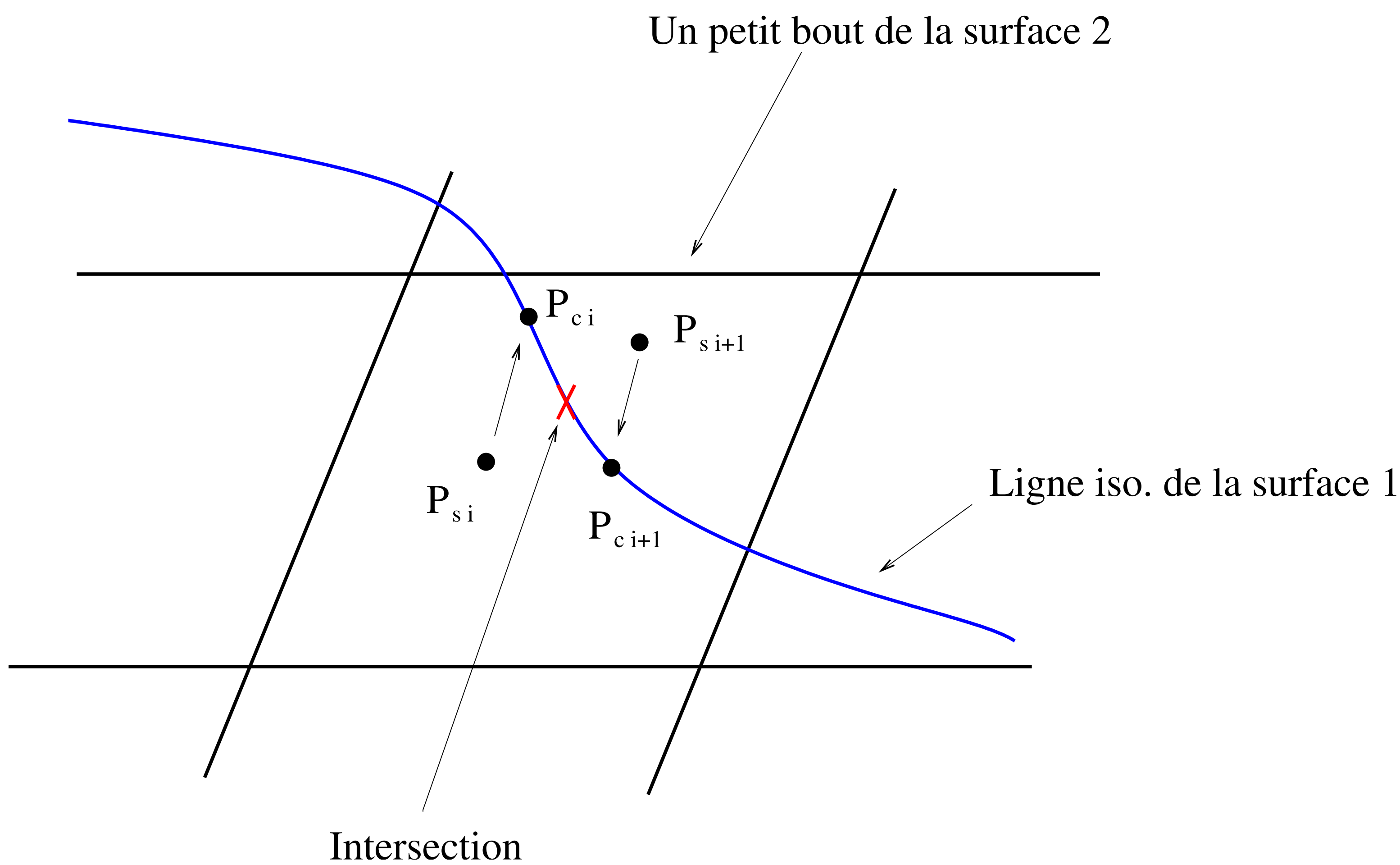
I - Identifier les points de départ

On discrétise les lignes isoparamétriques et on balaye chaque ligne en évaluant la distance d'un point P_c sur la ligne isoparamétrique avec sa projection sur l'autre surface (P_s).

$$\vec{R} = (\vec{P}_c - \vec{P}_s)$$

Pour effectuer efficacement ce type de projection, on peut avoir recours à des structures de données spécialisées comme un ADT, un quadtree, etc.

I - Identifier les points de départ



ÉCOLE POLYTECHNIQUE MONTRÉAL

I - Identifier les points de départ

$$
\begin{aligned}
\vec{R}_i &= (\vec{P}_{ci} - \vec{P}_{si}) \\
\vec{R}_{i+1} &= (\vec{P}_{ci+1} - \vec{P}_{si+1})
\end{aligned}
$$

Si $\vec{R}_i \cdot \vec{R}_{i+1} < 0$
$\Rightarrow$ Il y a intersection en quelque part entre $\vec{P}_{ci}$ et $\vec{P}_{ci+1}$ On résoud numériquement le problème $P_c - P_s = 0$ en utilisant comme point de départ $\frac{P_{ci}+P_{ci+1}}{2}$

Comme on est sur une ligne isoparamétrique u ou v, il reste 3 inconnues (s, t) et u ou v.

I - Identifier les points de départ

$$S_1(u, v) - S_2(s, t) = 0 \quad \text{en } x, y \text{ et } z.$$

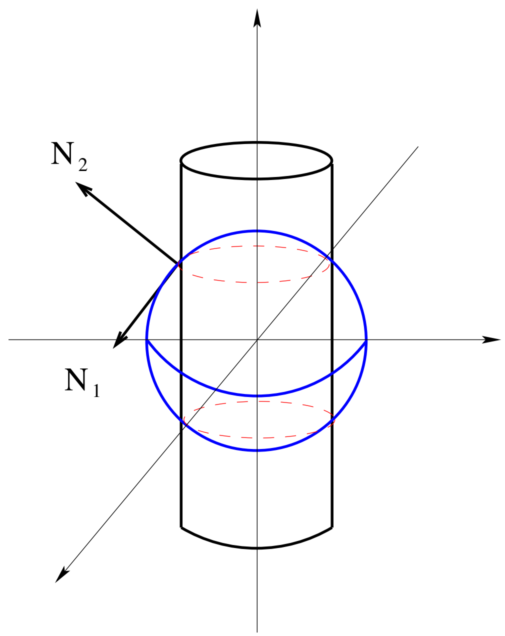
On a un bon point de départ donné par P_{ci} et P_{ci+1}, P_{si} et P_{si+1} et on limite u et v entre les valeurs (u_i, v_i) et (u_{i+1}, v_{i+1}). On obtient ainsi un point de départ d'une courbe d'intersection, que l'on stoke par exemple dans un tableau.

L'ordre des points de départ dans le tableau dépend de la paramétrisation de S_1 et de l'ordre de balayage des lignes isoparamétriques.

II - Retrouver les courbes d'intersection

En partant d'un point d'intersection, on construit la tangente à la courbe d'intersection en prenant :

$$\left(\frac{\partial S_1}{\partial u} \times \frac{\partial S_1}{\partial v}\right) \times \left(\frac{\partial S_2}{\partial s} \times \frac{\partial S_2}{\partial t}\right)$$



II - Retrouver les courbes d'intersection

On choisit la direction sur la tangente qui fait qu'en avançant sur la courbe, on va vers l'interieur de la cellule dans l'espace paramétrique de S_1.

On avance le long de la tangente d'un incrément $\Delta\sigma$

$u_{i+1} = u_i + (\frac{du}{d\sigma})\Delta\sigma \rightarrow$ calculé à partir des dérivées des 2 surfaces

$\frac{du}{d\sigma}$ est un incrément lié à la courbure de la courbe d'intersection, qui provient de paramètres locaux des deux surfaces.

II - Retrouver les courbes d'intersection

On construit une courbe d'intersection dans l'espace paramétrique de chaque surface.

Chaque fois que l'on fait un pas, on résoud

$$P_s - P_c = 0$$

en utilisant le nouveau point comme estimé initial.

On repart chaque fois du nouveau point calculé en avançant jusqu'à la frontière de la cellule.

Le chemin parcouru à l'intérieur de la cellule est conservé comme segment d'une des courbes d'intersection. En fonction de l'ordre de découverte des points initiaux, les segments sont trouvés dans un ordre quelconque.

III -Reconnecter les segments

Une fois que l'on a parcouru toutes les cellules, on doit reconnecter les segments et les orienter pour reconstruire les courbes complètes d'intersection. Pour reconnecter, on se base sur les points extrémités de chaque segment.

Il peut être nécessaire de reconnecter en 2 passes, dépendant de l'ordre dans lequel on trouve les intersections.

IV -Spliner les courbes paramétriques pour retrouver des courbes continues

Une fois que l'on a reconnecté les segments, on obtients deux courbes paramétriques linéaires par morceaux.

Pour chaque point, on a trois informations :

- les (u, v) de S_1
- les (s, t) de S_2
- les (x, y, z) proche à une tolérance donnée.

IV -Spliner les courbes paramétriques pour retrouver des courbes continues

Les trois points sont trois points qui peuvent être distinct, mais qui sont proche à une tolérance donnée.

On peut donc reconstruire trois courbes :

1. Espace géométrique en splinant les x, y, z
2. Espace paramétrique de S_1 en splinant les u, v
3. Espace paramétrique de S_2 en splinant les s, t

La courbe créée à partir des (x, y, z) est la courbe d'intersection "traditionnelle"