Chapitre III

Algorithmes de calcul de l'enveloppe convexe d'une polycourbe de Bézier

III.1. Introduction

Les résultats du chapitre précédent nous permettent d'affirmer que la suite des enveloppes convexes des polygones de contrôle successifs, obtenus par subdivision de de Casteljau, d'une polycourbe de Bézier converge, aussi bien au sens de la distance de Hausdorff qu'au sens topologique, et que la limite de cette suite est l'enveloppe convexe de la polycourbe de Bézier elle-même. Ces résultats nous permettent d'affirmer que des algorithmes calculant l'enveloppe convexe (topologique ou approchée) d'une polycourbe en utilisant la subdivision de de Casteljau convergent. Par contre, si dans le cas de l'enveloppe convexe approchée le test d'arrêt de l'algorithme est assez naturel, dans celui de la structure topologique de l'enveloppe convexe d'une polycourbe nous sommes loin d'avoir établi des critères d'arrêt. En effet, le théorème II.9 nous fournit un critère de pertinence topologique inutilisable algorithmiquement, car la valeur ε_0 définie dans sa preuve n'est pas calculable uniquement à partir du polygone de contrôle de la polycourbe. Nous devrons ainsi concevoir des conditions suffisantes pour la pertinence topologique de l'enveloppe convexe du polygone de contrôle pour celle de la polycourbe qui en plus doivent être facilement implémentables.

Nous commençons ce chapitre par la présentation de quelques types de données et algorithmes de base, dans la section III.3. Dans la section suivante, nous donnons quelques résultats, certains permettant de réduire les coûts de l'algorithme de calcul d'une approximation de l'enveloppe convexe d'une polycourbe et d'autres justifiant notre algorithme, et ensuite nous présentons cet algorithme en faisant aussi une discussion sur sa complexité. La cinquième section est consacrée au calcul de la structure topologique de l'enveloppe convexe d'une polycourbe et a la même composition que la quatrième (la précédente). La sixième section présente quelques considérations sur le calcul d'un polygone de contrôle fournissant en même temps une approximation de l'enveloppe convexe de sa polycourbe et la structure topologique de celle-ci, et les deux dernières sections contiennent des résultats numériques obtenus par l'implémentation des algorithmes présentés précédemment et respectivement quelques exemples.

III.2. Concentré de chapitre (III)

Dans ce chapitre, vous trouverez la méthode par laquelle, pour une polycourbe de Bézier donnée, peut être calculée une approximation, au sens de la distance de Hausdorff, de son enveloppe convexe, ou un polygone de contrôle fournissant la structure topologique de cette enveloppe convexe, ou même un polygone de contrôle qui en même temps fournit la structure topologique et une approximation de l'enveloppe convexe de la polycourbe. Je préviens le lecteur que l'implémentation de tous les algorithmes a été faite en C++, et par conséquent j'en ai plus ou moins emprunté le jargon pour décrire les structures de données et pour donner les schémas des algorithmes.

Ce chapitre est bien "touffu", et il est très possible qu'une lecture (presqu') intégrale vous parraîsse moins fatigante que des suivre les conseils de lecture "écourtée" - j'ai du mal moi-même à me suivre. En plus, dans les descriptions d'algorithme le chapitre a un peu la structure d'un fichier source C++, les procédures appelées par plusieurs programmes se trouvent au début, le plus facile étant alors de "sauter" la sous-section III.3.2 et d'y revenir à chaque fois qu'une procédure y étant décrite est utilisée.

Premier niveau. La section III.3 peut être complétement ignorée, mises à part les quelques premières phrases de chacun des paragraphes de la sous-section III.3.2, qui présentent brièvement les procédures qui seront appelées aussi bien dans le calcul d'une approximation de l'enveloppe convexe que dans celui de la structure topologique. Dans la sous-section III.4.1, les preuves des résultats n'ont pas à être lues, comme d'habitude, mais suivez bien les énoncés et les explications intercalées. Dans la sous-section III.4.2, le paragraphe III.4.2.1 est de moindre importance, sa première phrase expliquant ce que la procédure incriminée fait. Elle, avec les quelques autres de la section III.3, devrait vous permettre de suivre le fonctionnement de l'algorithme `Enveloppe_convexe_approchée`, paragraphe III.4.2.2. La sous-section III.4.3 est très importante pour comprendre les opérations executées par l'algorithme, et c'est une discussion utile pour la section III.5, où nous étudions la structure topologique de l'enveloppe convexe et où cette analyse ne sera plus faite. L'évaluation de coût se trouvant en fin de sous-section ne peut pas être suivie à ce niveau de lecture, mais la complexité totale est indiquée. Pour la section III.5, le procédé de lecture est identique : la section III.5.1 doit être lue en entier, mises à part les démonstrations (la notion, extrêmement importante, de dégénérescence, et celle apparentée de dégénérescence à ε-près sont introduites ici, paragraphes III.5.1.3 et III.5.1.4 - ne les ratez surtout pas) ; dans la sous-section III.5.2, lisez III.5.2.2 et le résultat obtenu en fin de III.5.2.6, plus III.5.2.7. Les sections III.6 à III.8 sont à lire intégralement.

Deuxième niveau. Mises à part les preuves (surtout celles de la sous-section III.5.1, qui sont pour la plupart des études de cas assez techniques), tout est à lire.

Avant de commencer, rappelons quelques conventions et notations couramment utilisées. Nous allons étudier, dans tous les cas, uniquement des polycourbes sans tangence intérieure, sans en faire la précision à chaque fois. En plus, nous considérerons toujours que le polygone de contrôle de la polycourbe est pseudo-simple. $\mathcal{B}$ désignera toujours une polycourbe de Bézier de polygone de contrôle $\mathcal{P}$. On notera $\mathcal{E}(\mathcal{P})$ l'enveloppe convexe du polygone de contrôle $\mathcal{P}$ et $\mathcal{Q} = \delta(\mathcal{E}(\mathcal{P}))$ le polygone frontière de $\mathcal{E}(\mathcal{P})$, et $\mathcal{E}(\mathcal{B})$ l'enveloppe convexe de la polycourbe et $\mathcal{E} = \delta(\mathcal{E}(\mathcal{B}))$ la courbe frontière de $\mathcal{E}(\mathcal{B})$. Comme nous avons vu dans le chapitre précédent, le polygone $\mathcal{Q}$ s'écrit

$$\mathcal{Q} = \mathrm{P}_{j_{i_1}}^{(i_1)}\mathrm{P}_{j_{i_1}+1}^{(i_1)}\ldots\mathrm{P}_{j_{i_1}+\ell_{i_1}}^{(i_1)}\mathrm{P}_{j_{i_2}}^{(i_2)}\ldots\mathrm{P}_{j_{i_p}+\ell_{i_p}}^{(i_p)},$$

avec $p \leq n$, $i_1, i_2, \ldots, i_p \in \{1, \ldots, n\}$ et $i_1 < i_2 < \ldots < i_p$.

III.3. Structures de données et algorithmes généraux

Dans cette section, nous présentons les structures de données et les algorithmes que nous allons utiliser aussi bien pour le calcul d'une enveloppe convexe approchée pour une polycourbe donnée que pour calculer la structure topologique de l'enveloppe convexe de la polycourbe. L'implémentation de ces algorithmes ayant été faite en C++, le vocabulaire que nous utiliserons est emprunté à ce langage. Nous précisons aussi que tous les tableaux sont indexés à partir de 0, allant ainsi jusqu'à $n-1$, si n est le nombre d'éléments du tableau.

III.3.1. *Structures de données*

La classe de base que nous avons créée est le `méta_polygone`, classe paramètrée par le type des membres de son champ `éléments`, et presque toutes les autres classes que nous présentons ici sont ses héritières. Cette classe a deux champs :
- `taille` : le nombre de membres du `méta_polygone` ;
- `éléments` : un tableau contenant les `taille` composantes du `méta_polygone`.

Cette classe est une généralisation (par paramétrisation par type) du modèle classique codant un polygone. Si $\mathcal{M}$ est un `méta-polygone`, $\mathcal{M}$.`éléments` désignera le tableau de ses composantes et pour un indice `i`$\in \{0, \ldots, \texttt{taille}-1\}$ $\mathcal{M}$.`éléments(i)` sera son `i`-ème composante. Pour simplifier, nous utiliserons $\mathcal{M}[\mathrm{i}]$ pour designer $\mathcal{M}$.`éléments(i)`.

La classe `polygone` code uniquement les polygones pour lesquels un intérieur peut être défini de manière cohérente (il n'y a pas que les polygones simples qui satisfont cette propriété) ; `polygone` n'est en fait qu'un autre nom donné à `méta_polygone<point>`, où le type `point` est une paire de réels représentant les coordonnées d'un point. Les sommets d'un polygone donnés dans le tableau de `points` du champ `éléments` sont ordonnés trigonométriquement.

La `courbe_de_Bézier` est une classe qui hérite la classe `polygone`, mais pour laquelle nous définissons en plus les champs suivants :
- `degré` : le degré de la courbe Bézier ;
- `rang` : l'indice que la courbe de Bézier de laquelle la courbe courante provient avait dans la structure initiale de la polycourbe ;
- `profondeur` : le nombre de subdivisions appliquées à la courbe $\mathcal{B}_{\mathrm{rang}}$ de la composition initiale de la polycourbe pour obtenir la courbe courante ;
- `intervalle` : une paire de réels définissant l'intervalle de paramètres auquel la courbe courante correspond par rapport à $\mathcal{B}_{\mathrm{rang}}$.

Le B-polygone d'une `courbe_de_Bézier` sera bien sur donné par le champ `éléments` que celle-ci hérite du `polygone`.
Les champs de la `courbe_de_Bézier` sont liés par certaines relations. Par exemple, `éléments` contient `degré`+1 =`taille` points et en considérant `intervalle`= (a, b) a et b seront de la forme

$$a = k/2^{\mathrm{profondeur}} \qquad b = (k+1)/2^{\mathrm{profondeur}}$$

où $0 \leq k \leq 2^{\mathrm{profondeur}} - 1$.

La classe codant une polycourbe de Bézier, nommée `polycourbe`, n'est pas une héritière de `méta_polygone`, mais une liste doublement chaînée de `courbes_de_Bézier` dans laquelle les courbes de la composition de la polycourbe sont ordonnées en sens trigonométrique. Ce choix est motivé par la nature des opérations les plus fréquemment effectuées sur une `polycourbe`, qui sont la recherche d'un de ses éléments et l'insertion de nouveaux éléments (suite aux subdivisions). Ainsi, la recherche s'effectue dans tous les cas “pas à pas”, c'est-à-dire que nous ne cherchons pas directement la i-ème courbe de Bézier de la polycourbe, mais la (les quelques) courbe(s) suivant - ou précédant - la courbe courante, donc la recherche se fait en temps constant aussi bien dans un tableau que dans la liste doublement chaînée. Par contre, l'insertion d'un nouveau élément se fait toujours en temps constant dans la liste doublement chaînée, mais en $\mathcal{O}(n-i)$ dans un tableau, où n est la taille du tableau et i la position où le nouvel élément est introduit.

La dernière classe que nous introduisons ici est la `liste_courbes`, le nom que nous utiliserons pour `méta_polygone<courbe_impliquée>`, chaque `courbe_impliquée` ayant deux champs :

- `ident_courbe` : un pointeur à la courbe de Bézier courante dans la structure de la polycourbe;
- `état_courbe` : un caractère, qui sera "n" si il existe des sommets de $\mathcal{Q}$ qui ne sont pas des sommets du polygone associé à `*ident_courbe` et au moins une des conditions suivantes est satisfaite, et "y" dans le cas contraire :
  - tous les sommets du polygone associé à `*ident_courbe` sont des sommets de $\mathcal{Q}$;
  - les polygones `*ident_courbe` et $\mathcal{Q}$ n'ont pas de sommet commun autre que les extrémités de `*ident_courbe`.

Nous verrons dans la sous-section III.4.1 que les courbes de Bézier pour lesquelles `courbe_impliquée.état_courbe` est "n" n'ont pas à être subdivisées dans le processus de construction de l'enveloppe convexe (approchée ou topologique) de la polycourbe.

III.3.2. *Algorithmes*

Nous donnons ici quelques algorithmes qui ne sont pas spécifiques au type d'enveloppe convexe que nous voulons calculer, étant ainsi nécessaires dans les deux sections suivantes de ce chapitre : l'algorithme de calcul de l'enveloppe convexe d'un polygone pseudo-simple, l'algorithme qui construit, en utilisant le précédent, l'enveloppe convexe du polygone de contrôle d'une polycourbe et finalement l'algorithme de subdivision d'une courbe de Bézier vue comme une arête d'une polycourbe. À la fin de chacun des paragraphes nous évaluons les coûts des algorithmes présentés.

III.3.2.1. L'enveloppe convexe d'un polygone pseudo-simple

La procédure que nous présentons ici, appelée `enveloppe_convexe_polygone`, est une variante de l'algorithme de Graham et Yao de calcul de l'enveloppe convexe d'un polygone simple [Gra83] qui est conçue pour fonctionner sur des polygones pseudo-simples et en plus rend en résultat un polygone avec le maximum de sommets possible.

Plus exactement, quand trois (ou plusieurs) sommets d'un polygone $\mathcal{P}$ sont colinéaires, bien ordonnés et se situent tous les trois sur la frontière de l'enveloppe convexe de $\mathcal{P}$, tous les trois seront des sommets de $\mathcal{Q}$, le polygone frontière de $\mathcal{E}(\mathcal{P})$. Une telle situation est présentée dans la figure III.1(a). La figure III.1(b) nous présente le cas de trois sommets qui sont colinéaires, mais ne sont pas correctement ordonnés et ne peuvent pas tous être des sommets de $\mathcal{Q}$. Nous avons vu dans la sous-section II.4.3 pourquoi il est nécessaire de construire un polygone $\mathcal{Q}$ maximal du point de vue du nombre de sommets.

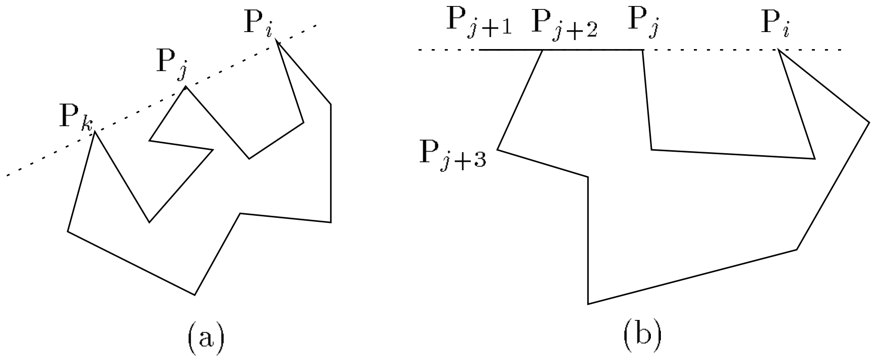

Figure III.1 (a) P_i, P_j, P_k *sont sommets de* $\mathcal{Q}$; (b) P_{j+2} *n'est pas sommet de* $\mathcal{Q}$

Nous supposons que l'abscisse du sommet $\mathcal{P}(0)$, le premier sommet de $\mathcal{P}$, est minimale dans l'ensemble des abscisses des sommets de $\mathcal{P}$. Avant tout appel de cette procédure par un autre algorithme nous nous assurerons que cette condition est satisfaite. Sans trop entrer dans les détails (démonstration de correction ou de convergence), nous présentons la structure de la procédure `enveloppe_convexe_polygone` :

1. `Enveloppe_convexe_polygone(polygone` $\mathcal{P}$, `ensemble_entiers K)`
2. `entier n=` nombre de sommets de $\mathcal{P}$
3. `entier i`
4. `begin`
5. `ajouter 0 à K`
6. `pour i allant de 1 à n-1`
7.   `begin`
8.   `si K a un seul élément`
9.   `ajouter i à K`
10.   `sinon`
11.   `begin`
12.   `tant que K a au moins deux éléments` *et*
$\mathcal{P}(\mathtt{i}) \notin \mathrm{DP}_{\mathrm{gauche}}(\mathcal{P}(\mathtt{K}_{\mathrm{avant-dernier}}), \mathcal{P}(\mathtt{K}_{\mathrm{dernier}}))$ *et*
$\mathcal{P}(\mathtt{K}_{\mathrm{dernier}}) \notin \,]\mathcal{P}(\mathtt{i}), \mathcal{P}(\mathtt{K}_{\mathrm{avant-dernier}})[$
13.   `éliminer dernier de K`
14.   `ajouter i à K`
15. $\mathtt{i} = \mathtt{i} + 1$
16.   `si`   `i est inférieur ou égal à n-1` *et*
$\mathcal{P}(\mathtt{i}) \in \mathrm{DP}_{\mathrm{gauche}}(\mathcal{P}(\mathtt{K}_{\mathrm{avant-dernier}}), \mathcal{P}(\mathtt{K}_{\mathrm{dernier}}))$ *et*
$\mathcal{P}(\mathtt{i}) \in \mathrm{DP}_{\mathrm{droit}}(\mathcal{P}(\mathtt{K}_{\mathrm{dernier}} - 1), \mathcal{P}(\mathtt{K}_{\mathrm{dernier}}))$
17.   `tant que i ne dépasse pas n-1` *et*
$\mathcal{P}(\mathtt{i}) \in \mathrm{DP}_{\mathrm{gauche}}(\mathcal{P}(\mathtt{K}_{\mathrm{avant-dernier}}), \mathcal{P}(\mathtt{K}_{\mathrm{dernier}}))$
18. $\mathtt{i} = \mathtt{i} + 1$
19.   `si i est inférieur ou égal à n-1)`
20.   `aller à la ligne 12`
21.   `end`
22.   `end`
23. `tant que` $\mathcal{P}(0) \notin \mathrm{DP}_{\mathrm{gauche}}(\mathcal{P}(\mathtt{K}_{\mathrm{avant-dernier}}), \mathcal{P}(\mathtt{K}_{\mathrm{dernier}}))$ *et*
$\mathcal{P}(\mathtt{K}_{\mathrm{dernier}}) \notin \,]\mathcal{P}(0), \mathcal{P}(\mathtt{K}_{\mathrm{avant-dernier}})[$

24. `éliminer dernier de K`
25. `end`

La troisième condition de la ligne 12 (qui est la même que la deuxième condition de la ligne 23, mais pour d'autres indices des sommets concernés) est celle qui confère à l'algorithme les propriétés annoncées. Le polygone $\mathcal{Q}$ est identifié par l'ensemble `K` des indices que les sommets respectifs de $\mathcal{Q}$ ont dans l'écriture du polygone $\mathcal{P}$. Il est très facile de voir que cet algorithme est linéaire en le nombre de sommets du polygone pour lequel nous voulons calculer l'enveloppe convexe.

III.3.2.2. L'enveloppe convexe du polygone de contrôle

En utilisant la procédure présentée dans le paragraphe précédent, cet algorithme calcule, pour une polycourbe codée en un objet de type `polycourbe`, l'enveloppe convexe du polygone de contrôle. Ce qui nous intéressera dans les sections suivantes ne sera en fait pas le polygone $\mathcal{Q}$. Ce polygone nous sert uniquement pour identifier les courbes de Bézier qui doivent être subdivisées, et donc la procédure suivante aura comme résultat une `liste_courbes`. Le schéma de l'algorithme est le suivant :

1. `Enveloppe_convexe_polygone_contrôle(polycourbe` $\mathcal{B}$, `liste_courbes` $\mathcal{L}$)
2. `polygone` $\mathcal{P}$, $\mathcal{P}_1$
3. `ensemble_entiers I, K`
4. `entier minim`
5. `begin`
6. `polygone_de_contrôle(`$\mathcal{B}$, $\mathcal{P}$`)`
7. `abscisse_minimale(`$\mathcal{P}$, `minim)`
8. `renumérotage_polygone(`$\mathcal{P}$, `minim`, $\mathcal{P}_1$`)`
9. `enveloppe_convexe_polygone(`$\mathcal{P}_1$, `K)`
10. `renumérotage(K, minim, I)`
11. `identification_courbes(I,` $\mathcal{B}$, $\mathcal{L}$`)`
12. `end`

La procédure `polygone_de_contrôle`, de la ligne 6, construit le polygone de contrôle d'une polycourbe donnée, en concatenant les B-polygones donnés par le champ `éléments` de chacune des `courbes_de_Bézier` de la `polycourbe`. La concatenation est faite tel que les sommets extrémités des courbes de Bézier n'apparaîssent pas deux fois dans le tableau `polygone.éléments`. La complexité de cet algorithme est $\mathcal{O}(M)$, où $M = \sum_{i=1}^{n} m_i$ est le nombre de sommets du polygone de contrôle de la polycourbe, il y a $\mathcal{O}(1)$ opérations à exécuter pour chaque sommet de $\mathcal{P}$.

La procédure `abscisse_minimale` retourne, pour un polygone donné, l'indice que le sommet d'abscisse minimale a dans la liste de sommets du polygone. Si plusieurs sommets ont la même abscisse, qui est l'abscisse minimale des tous les sommets du polygone, alors la procédure retourne le plus petit des indices des ces sommets. Cet algorithme aussi est linéaire en M, c'est l'algorithme classique de calcul du plus petit élément d'un vecteur.

La procédure `renumérotage_polygone` construit, à partir du polygone $\mathcal{P}$ et de l'entier `minim` (qui est l'indice d'un des sommets de $\mathcal{P}$), le polygone $\mathcal{P}_1$ qui a les mêmes sommets que $\mathcal{P}$, mais avec une rotation des indices telle que le sommet qui avait l'indice `minim` dans le polygone $\mathcal{P}$ ait l'indice 0 dans le polygone $\mathcal{P}_1$. La complexité de la procédure est toujours $\mathcal{O}(M)$, chacun des sommets est manipulé une seule fois.

La procédure `renumérotage` recupère, à partir de l'ensemble `K` et de l'entier `minim`, l'ensemble `I` des indices des sommets de $\mathcal{P}$ qui sont des sommets de $\mathcal{Q}$ (rappellons que l'ensemble `K` contient les indices des sommets du polygone $\mathcal{P}_1$ qui sont des sommets de $\mathcal{Q}$, il faut donc retrouver la correspondance entre les sommets de $\mathcal{P}_1$ et ceux de $\mathcal{P}$). L'ensemble `I` sera ensuite ordonné, la sortie de cette procédure étant ainsi une suite strictement croissante d'indices de sommets du polygone $\mathcal{P}$. La complexité de cette procédure est $\mathcal{O}(\text{cardinal}(\texttt{K}))$, et nous aurons toujours $\text{cardinal}(\texttt{K}) \leq M$.

Enfin, la dernière procédure de `enveloppe_convexe_polygone_contrôle`, `identification_courbes`, construit, à partir de l'ensemble d'entiers `I` et de $\mathcal{B}$, la `liste_courbes` $\mathcal{L}$ qui indique les courbes de Bézier de la composition de la polycourbe qui sont susceptibles de devoir être subdivisées. Cette construction se fait comme la définition de la classe `liste_courbes` l'indique : toute courbe de Bézier dont au moins un point de contrôle est un sommet de $\mathcal{Q}$ apparaîtra dans $\mathcal{L}$, et à chacune de ces courbes un caractère, "n" ou "y", sera associé comme décrit dans la sous-section III.2.1. Cette procédure est linéaire en le nombre d'éléments de l'ensemble `I`, qui est inférieur ou égal à M.

Nous pouvons maintenant voir que la complexité de la procédure `enveloppe_convexe_polygone_contrôle` est $\mathcal{O}(M)$, car chacune des procédures qu'elle appelle, `enveloppe_convexe_polygone` comprise, a cette complexité.

III.3.2.3. La subdivision d'une courbe de Bézier

Nous allons présenter dans ce paragraphe une procédure un peu plus complexe que ne le serait une calculant uniquement les deux `courbes_de_Bézier` obtenues par la subdivision d'une `courbe_de_Bézier` donnée. La raison est que nous n'aurons jamais besoin de subdiviser une courbe de Bézier isolée. Nous aurons besoin de calculer, pour une polycourbe $\mathcal{B}$ ayant une structure courante $\mathcal{B} = \cup_{i=1}^{n} \mathcal{B}_i$, la structure obtenue après la subdivision de certaines des courbes de Bézier de sa composition. Les courbes à subdiviser seront identifiées par un tableau de pointeurs.

Le schéma de la procédure `subdivision` est extrêmement simple :

```
Subdivision(polycourbe 𝓑, tableau_pointeurs J)
courbe_de_Bézier 𝓑', 𝓑''
entier k, i
begin
k = nombre d'éléments de J
pour i allant de 0 à k-1
     begin
     subdiviser(*J(i), 𝓑', 𝓑'')
     remplacer *J(i) par 𝓑' dans 𝓑
     introduire 𝓑'' dans 𝓑 juste après 𝓑'
     end
end
```

Cette procédure modifie la structure de la `polycourbe` $\mathcal{B}$, en subdivisant toutes les courbes de Bézier identifiées par le `tableau_pointeurs J` et en remplaçant chacune des courbes à subdiviser par les deux courbes obtenues après subdivision. Elle fait appel à une seule autre procédure, `subdiviser`. Cette procédure prend comme paramètre une `courbe_de_Bézier` $\mathcal{B}$ et retourne deux `courbe_de_Bézier` $\mathcal{B}'$ et $\mathcal{B}''$, dont les champs ont les valeurs suivantes :

- $\mathcal{B}'$.`taille` = $\mathcal{B}''$.`taille` = $\mathcal{B}$.`taille` ;
- $\mathcal{B}'$.`degré` = $\mathcal{B}''$.`degré` = $\mathcal{B}$.`degré` ;
- $\mathcal{B}'$.`rang` = $\mathcal{B}''$.`rang` = $\mathcal{B}$.`rang` ;
- $\mathcal{B}'$.`profondeur` = $\mathcal{B}'$.`profondeur` = $\mathcal{B}$.`profondeur` + 1 ;
- si $\mathcal{B}$.`interval` est donné par la paire de réels (a, b), alors $\mathcal{B}'$.`interval` sera donné par $(a, (a+b)/2)$ et $\mathcal{B}''$.`interval` sera donné par $((a+b)/2, b)$;
- si $\mathcal{B}$.`éléments` = $\mathrm{P}_0\mathrm{P}_1 \ldots \mathrm{P}_m$, alors $\mathcal{B}'$.`éléments` = $\mathrm{P}'_0\mathrm{P}'_1 \ldots \mathrm{P}'_m$ et $\mathcal{B}''$.`éléments` = $\mathrm{P}''_0\mathrm{P}''_1 \ldots \mathrm{P}''_m$, où

$$\mathrm{P}'_i = \frac{1}{2^i}\sum_{k=0}^{i} \mathrm{C}_i^k \mathrm{P}_k \qquad \mathrm{P}''_i = \frac{1}{2^{n-i}}\sum_{k=0}^{n-i} \mathrm{C}_{n-i}^k \mathrm{P}_{i+k} \quad \forall i \in \{0, 1, \ldots, m\}$$

Pour calculer la complexité de cette procédure, il faut étudier le coût du calcul des sommets des polygones de contrôle des deux courbes de Bézier obtenues après subdivision, car il est évident que le calcul des autres champs des `courbes_de_Bézier` $\mathcal{B}'$ et $\mathcal{B}''$ se fait en $\mathcal{O}(1)$. Si nous supposons que les valeurs des combinaisons C_i^k sont connues pour tout $i \in \{0, \ldots, m\}$, où m est le degré de la courbe de Bézier à subdiviser, et pour tout $k \in \{0, \ldots, i\}$, alors le calcul des points P'_i et P''_i, pour $i \in \{0, \ldots, m\}$, se fait en $\mathcal{O}(m(m+1)/2)$, ou plus grossièrement en $\mathcal{O}(m^2)$. Si elles ne sont pas connues, les calculer coûte aussi $\mathcal{O}(m(m+1)/2)$ et si l'algorithme ne les stocke pas et les recalcule à chaque fois, le coût total de l'algorithme devient $\mathcal{O}(m^4)$. D'un autre côté, le stockage de toutes ces valeurs prendra, dans la mémoire, $\mathcal{O}(m(m+1)/2)$ entiers. Nous pouvons donc adapter la façon de calculer les points P'_i et P''_i, pour $i \in \{0, \ldots, m\}$, aux caractéristiques de l'ordinateur sur lequel nous voulons implémenter l'algorithme. En plus, le coût du calcul des combinaisons peut être réduit de manière significative en utilisant des formules de récurrence. Nous allons donc considerer que la complexité de cette procédure est $\mathcal{O}(m^2)$. Mais si pour le calcul des points de contrôle des courbes $\mathcal{B}'$ et $\mathcal{B}''$ nous n'utilisons pas les formules ci-dessus, mais le tableau de la subdivision de de Casteljau (sous-section I.7.3), le coût sera $\mathcal{O}(m^2)$ sans que de l'espace mémoire supplémentaire dûsse être aloué. C'est cette dernière solution que nous avons adoptée.

Pour évaluer le coût de la procédure `subdivision`, nous devons regarder les coût des opérations effectuées aux lignes 9 et 10 et il est facile à voir, que le choix des structures de données que nous avons fait rend les deux instructions exécutables en temps constant à chaque itération du cycle des lignes 6-11. Par conséquent, le coût global de l'exécution de ces deux lignes, sommé pour toutes les itérations du cycle, est $\mathcal{O}(\mathrm{card}(\mathtt{J}))$, donc dans le cas le pire $\mathcal{O}(n)$.

La complexité totale de l'algorithme sera (dans le cas le pire), en sommant les coûts des opérations effectuées, $\mathcal{O}(nd^2)$, où $d = \max_{i=1}^{n} m_i$.

III.4. L'enveloppe convexe approchée

Dans cette section, nous présentons un algorithme calculant une approximation à ε-près (au sens de la distance de Hausdorff) de l'enveloppe convexe d'une polycourbe donnée, pour un ε donné lui aussi par l'utilisateur. Nous commençons par énoncer et prouver les résultats qui justifient notre algorithme, ensuite nous décrivons cet algorithme et dans la dernière sous-section nous évaluons sa complexité.

III.4.1. *Résultats théoriques*

Soit $\mathcal{B} = \cup_{i=1}^{n} \mathcal{B}_i$ une polycourbe et $\varepsilon > 0$. Supposons que l'on veut calculer une approximation de $\mathcal{E}(\mathcal{B})$ avec une erreur inférieure à ε, au sens de la distance de Hausdorff. Nous allons avoir besoin de connaître , pour tout $i \in \{1, \dots, n\}$, la distance de Hausdorff entre $\mathcal{B}_i$ et $\mathcal{P}_i$, $\delta^{\mathrm{H}}(\mathcal{B}_i, \mathcal{P}_i)$. Comme la distance de Hausdorff entre une courbe de Bézier et son polygone de contrôle n'est pas calculable algorithmiquement, nous allons utiliser la majoration donnée par le lemme suivant.

Lemme III.1. – *Soit $\mathcal{B}_0$ une courbe de Bézier complètement convexe de polygone de contrôle $\mathcal{P}_0 = \mathrm{P}_0\mathrm{P}_1 \dots \mathrm{P}_m$. Alors*

$$\delta^{\mathrm{H}}(\mathcal{B}_0, \mathcal{P}_0) \leq \delta^{\mathrm{H}}([\mathrm{P}_0\mathrm{P}_m], \mathcal{P}_0)$$

Preuve – Pour démontrer cette inégalité, nous allons utiliser pour la distance de Hausdorff la formule (I.4) et prouver les inclusions

$$\mathcal{B}_0 \subset \mathrm{B}(\mathcal{P}_0, R) \quad \text{et} \quad \mathcal{P}_0 \subset \mathrm{B}(\mathcal{B}_0, R),$$

où $R = \delta^{\mathrm{H}}([\mathrm{P}_0\mathrm{P}_m], \mathcal{P}_0)$. Nous ferons, pour les deux inclusions, une preuve par l'absurde.

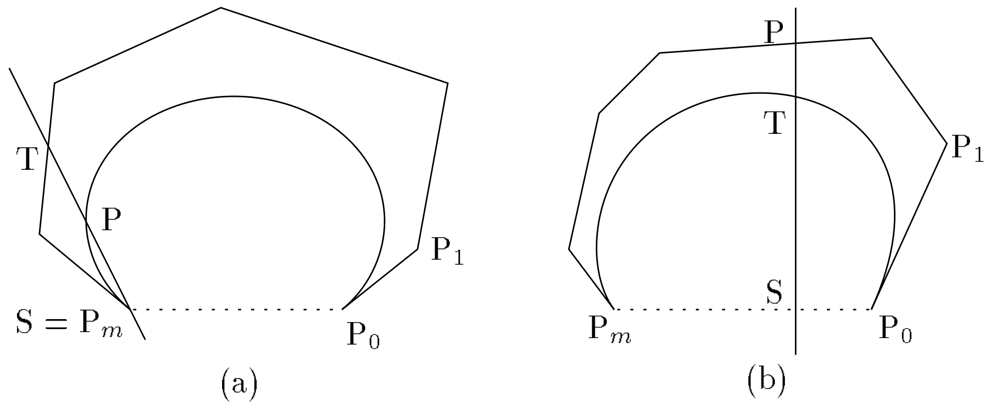

Figure III.2

Supposons que $\mathcal{B}_0 \not\subset \mathrm{B}(\mathcal{P}_0, R)$ et soit $\mathrm{P} \in \mathcal{B}_0$ tel que $\mathrm{P} \notin \mathrm{B}(\mathcal{P}_0, R)$. Évidemment, le point P ne peut pas être P_0 ou P_m. Nous notons S un point de $[\mathrm{P}_0\mathrm{P}_m]$ qui satisfait $\mathrm{d}(\mathrm{P}, [\mathrm{P}_0\mathrm{P}_m]) = \mathrm{d}(\mathrm{P}, \mathrm{S})$ et soit $\mathrm{T} = \mathrm{PS} \cap \left(\mathcal{P}_0 \setminus \{\mathrm{P}_0, \mathrm{P}_m\}\right)$ (figure III.2(a)). La courbe de Bézier étant complètement convexe, il est évident que le point T est unique et $\mathrm{P} \in]\mathrm{TS}[$. Il est alors facile à voir que $\mathrm{d}(\mathrm{T}, [\mathrm{P}_0\mathrm{P}_m]) = \mathrm{d}(\mathrm{T}, \mathrm{S})$; en effet, si nous supposons qu'il existe

$S' \in [P_0P_m]$ tel que $d(T,S') < d(T,S)$, soit $P' \in [P_0P_m]$ satisfaisant $PP' \parallel TS'$ (figure III.3).

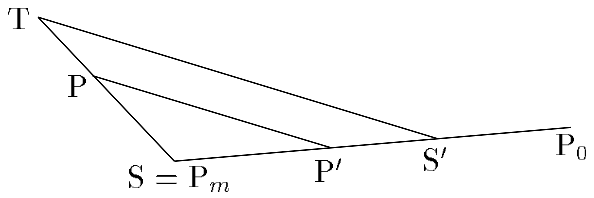

Figure III.3

Alors les triangles PP′S et TS′S sont semblables et nous avons

$$\frac{d(T,S')}{d(P,P')} = \frac{d(T,S)}{d(P,S)}, \quad \text{donc} \quad \frac{d(P,P')}{d(P,S)} = \frac{d(T,S')}{d(T,S)} < 1$$

d'où une contradiction avec le choix du point S (ceci est un cas particulier du résultat de [McM71] p31).

Mais le point T doit appartenir à $B([P_0P_m], R)$, de par la définition de R, et comme S est le point de $[P_0P_m]$ le plus près de lui, nous obtenons $d(T,S) \leq R$ et par conséquent $d(P,T) \leq R$. Nous avons ainsi trouvé un point $T \in \mathcal{P}_0$ tel que $P \in B(T,R)$, d'où la contradiction avec l'hypothèse $P \notin B(\mathcal{P}_0, R)$.

Pour démontrer la deuxième inclusion, supposons qu'il existe un point $P \in \mathcal{P}_0$ tel que $P \notin B(\mathcal{B}_0, R)$. Remarquons cette fois aussi que $P \notin \{P_0, P_m\}$ et soit $S \in [P_0P_m]$ tel que $d(P, [P_0P_m]) = d(P,S)$. Nous notons $T = PS \cap \big(\mathcal{B}_0 \setminus \{P_0, P_m\}\big)$ (figure III.2(b)) et évidemment $T \in]PS[$. De la même manière, nous avons $d(P,S) \leq R$ et donc $d(P,T) \leq R$, contradiction avec l'hypothèse $P \notin B(\mathcal{B}_0, R)$. □

La méthode "bête et méchante" de calcul de l'approximation voulue serait de subdiviser toutes les courbes $\mathcal{B}_i$ de la composition de la polycourbe jusqu'au moment où $\delta^H([P_0^{(i)}P_{m_i}^{(i)}], \mathcal{P}_i) < \varepsilon$ et ensuite construire l'enveloppe convexe du polygone de contrôle obtenu. Cette méthode est justifiée par les lemmes II.1 et II.3, qui impliquent que l'enveloppe convexe ainsi obtenue est une approximation à ε-près de l'enveloppe convexe de la polycourbe. Bien sur, cet algorithme est loin d'être optimal et nous cherchons des critères pour réduire le nombre de courbes de Bézier à subdiviser. Le lemme suivant nous fournit un tel critère.

Lemme III.2. – *Soit $\mathcal{I} = \{i_1, i_2, \ldots, i_p\}$ tel que le polygone frontière de $\mathcal{E}(\mathcal{P})$, $\mathcal{Q}$, s'écrive sous la forme de* (II.2) *et soient*

$$\mathcal{I}_1 = \big\{ i \in \mathcal{I} \mid \ell_1 = 0 \text{ et } P_{j_i}^{(i)} \in \{P_0^{(i)}, P_{m_i}^{(i)}\} \big\}$$

$$\mathcal{I}_2 = \big\{ i \in \mathcal{I} \mid \ell_1 = 1 \text{ et } P_{j_i}^{(i)} = P_0^{(i)}, P_{j_i+\ell_i}^{(i)} = P_{m_i}^{(i)} \big\}$$

Nous posons $\mathcal{I}_0 = \mathcal{I} \setminus (\mathcal{I}_1 \cup \mathcal{I}_2)$ et alors

$$\delta^H\big(\mathcal{E}(\mathcal{B}), \mathcal{E}(\mathcal{P})\big) \leq \max_{i \in \mathcal{I}_0} \delta^H(\mathcal{B}_i, \mathcal{P}_i)$$

Preuve – Pour démontrer l'inégalité de l'énoncé, nous devons prouver les deux inclusions suivantes :

$$\mathcal{E}(\mathcal{B}) \subset \mathrm{B}\big(\mathcal{E}(\mathcal{P}), \max_{i\in\mathcal{I}_0} \delta^{\mathrm{H}}(\mathcal{B}_i, \mathcal{P}_i)\big) \qquad \mathcal{E}(\mathcal{P}) \subset \mathrm{B}\big(\mathcal{E}(\mathcal{B}), \max_{i\in\mathcal{I}_0} \delta^{\mathrm{H}}(\mathcal{B}_i, \mathcal{P}_i)\big)$$

Mais nous savons $\mathcal{E}(\mathcal{B}) \subset \mathcal{E}(\mathcal{P})$, donc la première inclusion est vérifiée et il nous reste à prouver que la deuxième l'est aussi. Dans ce but, nous remarquons que l'enveloppe convexe de $\mathcal{P}$ est égale à l'enveloppe convexe des sommets du polygone $\mathcal{Q}$, nous pouvons ainsi écrire $\mathcal{E}(\mathcal{P}) = \mathcal{E}(\mathcal{A})$, où

$$\mathcal{A} = \bigcup_{i\in\mathcal{I}_0} \mathcal{P}_i \cup \big\{\mathrm{P}_0^{(i)} \mid i \in \mathcal{I}_1 \cup \mathcal{I}_2\big\} \cup \big\{\mathrm{P}_{m_i}^{(i)} \mid i \in \mathcal{I}_1 \cup \mathcal{I}_2\big\},$$

car $\mathrm{P}_{j_{i_k}}^{(i_k)}, \mathrm{P}_{j_{i_k}+1}^{(i_k)}, \ldots, \mathrm{P}_{j_{i_k}+\ell_{i_k}}^{(i_k)} \in \mathcal{A}$ pour tout $k \in \{1, \ldots, p\}$.

Il est facile à voir que la boule de centre un ensemble convexe est un ensemble convexe, et ainsi il nous reste à prouver

$$\mathcal{A} \subset \mathrm{B}\big(\mathcal{E}(\mathcal{B}), \max_{i\in\mathcal{I}_0} \delta^{\mathrm{H}}(\mathcal{B}_i, \mathcal{P}_i)\big),$$

car $\mathcal{E}(\mathcal{B})$ est convexe, donc $\mathrm{B}\big(\mathcal{E}(\mathcal{B}), \max_{i\in\mathcal{I}_0} \delta^{\mathrm{H}}(\mathcal{B}_i, \mathcal{P}_i)\big)$ est convexe et alors l'enveloppe convexe de $\mathcal{A}$ sera incluse dans cette boule.

Les points $\mathrm{P}_0^{(i)}$ et $\mathrm{P}_{m_i}^{(i)}$ sont des points de $\mathcal{B}$ pour tout $i \in \mathcal{I}_1 \cup \mathcal{I}_2$. D'un autre côté, si nous considérons $i \in \mathcal{I}_0$ et $\mathrm{P} \in \mathcal{P}_i$, alors

$$\mathrm{d}(\mathrm{P}, \mathcal{E}(\mathcal{B})) \leq \mathrm{d}(\mathrm{P}, \mathcal{B}) \leq \mathrm{d}(\mathrm{P}, \mathcal{B}_i) \leq \delta^{\mathrm{H}}(\mathcal{P}_i, \mathcal{B}_i)$$

et donc nous aurons

$$\bigcup_{i\in\mathcal{I}_0} \mathcal{P}_i \subset \mathrm{B}\big(\mathcal{E}(\mathcal{B}), \max_{i\in\mathcal{I}_0} \delta^{\mathrm{H}}(\mathcal{B}_i, \mathcal{P}_i)\big)$$

Ainsi, $\mathcal{A}$ est inclus dans la boule centrée en $\mathcal{E}(\mathcal{B})$ et de rayon $\max_{i\in\mathcal{I}_0} \delta^{\mathrm{H}}(\mathcal{B}_i, \mathcal{P}_i)$ et la preuve du lemme est finie. □

Ce lemme nous permet de subdiviser uniquement les courbes $\mathcal{B}_i$ avec $i \in \mathcal{I}_0$, donc celles dont le polygone de contrôle participe à la frontière de l'enveloppe convexe de $\mathcal{P}$ avec des sommets autres que les extrémités $\mathrm{P}_0^{(i)}$ et $\mathrm{P}_{m_i}^{(i)}$. Mais même parmi les courbes décrites ici il existe certaines qui ne doivent pas être subdivisées, un autre lemme justifiant leur élimination de la liste de courbes à subdiviser.

Lemme III.3. – *Soit $i \in \{1, \ldots, n\}$ tel que tous les sommets de $\mathcal{P}_i$ soient des sommets de $\mathcal{Q}$ et supposons que $\cup_{j\neq i}\mathcal{B}_j \not\subset reg(\mathcal{B}_i, \mathcal{P}_i) \cup \mathcal{P}_i$. Alors $\mathcal{B}_i \subset \mathcal{E} = \delta(\mathcal{E}(\mathcal{B}))$.*

Preuve – Si la courbe de Bézier $\mathcal{B}_i$ est un segment, l'affirmation de l'énoncé est évidemment vraie, car nous aurons $\mathcal{B}_i \subset \mathcal{Q}$ et comme $\mathcal{E}(\mathcal{B}) \subset \mathcal{E}(\mathcal{P})$ il en résulte tout de suite $\mathcal{B}_i \subset \mathcal{E}$. Supposons donc que $\mathcal{B}_i$ n'est pas un segment et soit

$$\mathcal{R}_i = \mathrm{DP}\big(\mathrm{P}_0^{(i)}\mathrm{P}_1^{(i)}, \mathcal{B}_i\big) \cap \mathrm{DP}\big(\mathrm{P}_{m_i}^{(i)}\mathrm{P}_{m_i-1}^{(i)}, \mathcal{B}_i\big) \cap \left(\bigcup_{k=1}^{m_i-2} \mathrm{DP}\big(\mathrm{P}_k^{(i)}\mathrm{P}_{k+1}^{(i)}, \mathcal{B}_i\big)\right)$$

Cette région est convexe, étant une intersection de demi-plans, non-vide, car le polygone $\mathcal{P}_i$ est convexe, et elle peut être bornée ou non, comme nous pouvons le voire figure III.4.

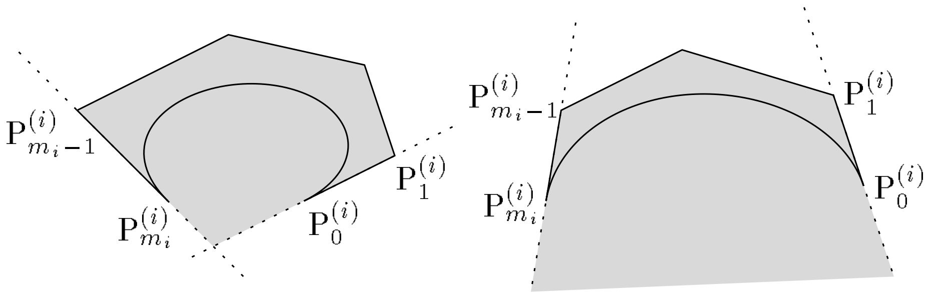

Figure III.4

L'hypothèse du lemme implique $\mathcal{E}(\mathcal{P}) \subset \overline{\mathcal{R}}_i$, donc $\mathcal{E}(\mathcal{B}) \subset \overline{\mathcal{R}}_i$ et donc $\mathcal{B} \subset \overline{\mathcal{R}}_i$. La polycourbe étant une courbe simple, nous en déduisons

$$\mathcal{B} \setminus \mathcal{B}_i \subset \overline{reg(\mathcal{B}_i, \mathcal{P}_i)} \setminus \mathcal{B}_i \quad \text{ou} \quad \mathcal{B} \setminus \mathcal{B}_i \subset \overline{\mathcal{R}}_i \setminus \overline{reg(\mathcal{B}_i, \mathcal{P}_i)}$$

La première de ces deux inclusions est exclue par hypothèse. Par conséquent, nous aurons $\mathcal{B} \setminus \mathcal{B}_i \subset \overline{\mathcal{R}}_i \setminus \overline{reg(\mathcal{B}_i, \mathcal{P}_i)}$ et nous en déduisons tout de suite l'inclusion

$$\mathcal{B} \subset \overline{\mathcal{R}}_i \setminus \big(reg(\mathcal{B}_i, \mathcal{P}_i) \cup \mathcal{B}_i\big)$$

En notant $\Delta_{\mathcal{B}_i}(t)$ la droite tangente à $\mathcal{B}_i$ en $\mathrm{P}_t = \mathrm{B}_{m_i}\big(\mathrm{P}^{(i)}_0, \ldots, \mathrm{P}^{(i)}_{m_i} \,;\, t\big)$, une conséquence directe des propriétés des courbes de Bézier et de la convexité de $\mathcal{B}_i$ sera l'égalité

$$\overline{\mathcal{R}}_i \setminus \big(reg(\mathcal{B}_i, \mathcal{P}_i) \cup \mathcal{P}_i\big) = \bigcap_{t\in[0,1]} \overline{\mathrm{DP}\big(\Delta_{\mathcal{B}_i}(t), \mathcal{B}_i\big)}$$

et le fait que cet ensemble est convexe. Comme la polycourbe $\mathcal{B}$ est incluse dans cet ensemble, son enveloppe convexe le sera aussi. Mais la courbe $\mathcal{B}_i$ est complètement incluse dans la frontière de $\overline{\mathcal{R}}_i \setminus \big(reg(\mathcal{B}_i, \mathcal{P}_i) \cup \mathcal{P}_i\big)$ et donc nous aurons $\mathcal{B}_i \subset \mathcal{E}$. □

La conséquence de ce résultat est que, après avoir une fois subdivisé toutes les courbes de Bézier de la composition de la polycourbe identifiées par le lemme III.2, nous pouvons ne plus subdiviser celles d'entre elles dont le polygone de contrôle a tous les sommets sur la frontière $\mathcal{Q}$ de l'enveloppe convexe du polygone de contrôle de la polycourbe. En effet, une démonstration très semblable à celle du lemme III.2 nous permet de prouver que, si on considère $\mathcal{AE}$ la courbe obtenue à partir du polygone $\mathcal{Q}$ en remplaçant le polygone $\mathcal{P}_i$, à chaque fois que tous ses sommets sont des sommets de $\mathcal{Q}$, par la courbe de Bézier correspondante $\mathcal{B}_i$ et on note $\mathcal{R}$ la région bornée par $\mathcal{AE}$, alors

$$\delta^{\mathrm{H}}(\mathcal{E}(\mathcal{B}), \mathcal{R}) \leq \max_{i\in\widetilde{\mathcal{I}}_0} \delta^{\mathrm{H}}(\mathcal{B}_i, \mathcal{P}_i),$$

où

$$\widetilde{\mathcal{I}}_0 = \big\{i \in \mathcal{I}_0 \mid \exists k \in \{0, \ldots, m_i\} \text{ tel que } \mathrm{P}^{(i)}_k \text{ n'est pas un sommet de } \mathcal{Q}\big\} \tag{III.1}$$

.

La condition indispensable pour prouver le lemme est

$$\mathcal{B} \setminus \mathcal{B}_i \cap \overline{reg(\mathcal{B}_i, \mathcal{P}_i)} \setminus \mathcal{B}_i = \emptyset$$

et nous cherchons une condition nécessaire pour que les deux ensembles ci-dessus soient disjoints. Comme nous supposons que le polygone de contrôle de la polycourbe est pseudo-simple, la seule configuration possible dans la quelle cette condition n'est pas satisfaite est présentée dans la figure III.5, où $\mathcal{B}_{i-1}$ et $\mathcal{B}_{i+1}$ sont des segments..

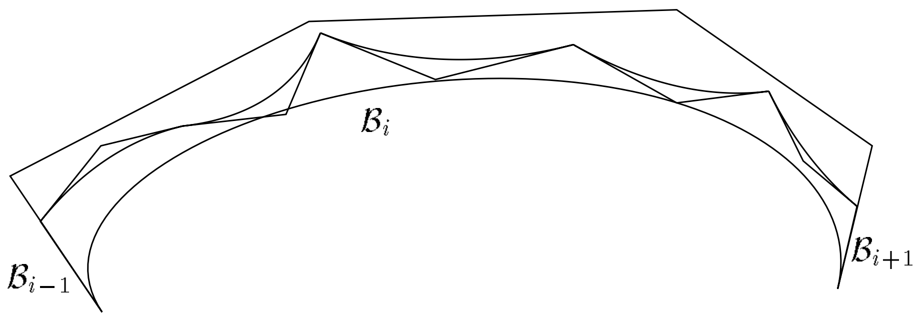

Figure III.5

Nous avons vu dans le paragraphe III.2.2.1 comment fonctionne la procédure que nous utiliserons pour calculer l'enveloppe convexe de $\mathcal{P}$. Ainsi, nous saurons que pour la polycourbe de la figure III.5 le polygone $\mathcal{Q}$ est égale au polygone $\mathcal{P}_i$ et, plus généralement, $\mathcal{Q} = \mathcal{P}_i$ si et seulement si le polygone $\mathcal{P}$ est contenu dans la zone fermée délimitée par le polygone $\mathcal{P}_i$ sans le segment $]\mathrm{P}_0^{(i)}\mathrm{P}_{m_i}^{(i)}[$, $\overline{reg(\mathcal{P}_i)}\backslash]\mathrm{P}_0^{(i)}\mathrm{P}_{m_i}^{(i)}[$. Donc la condition à imposer à un polygone de contrôle $\mathcal{P}_i$ dont les sommets sont tous des sommets de $\mathcal{Q}$ pour que la courbe de Bézier correspondante ne soit pas subdivisée sera simplement que le polygone $\mathcal{Q}$ ait d'autres sommets que ceux de $\mathcal{P}_i$. Si cette condition n'est pas satisfaite, nous devons subdiviser la courbe $\mathcal{B}_i$, pour trancher entre la situation de la figure III.5 et celle de la figure III.6.

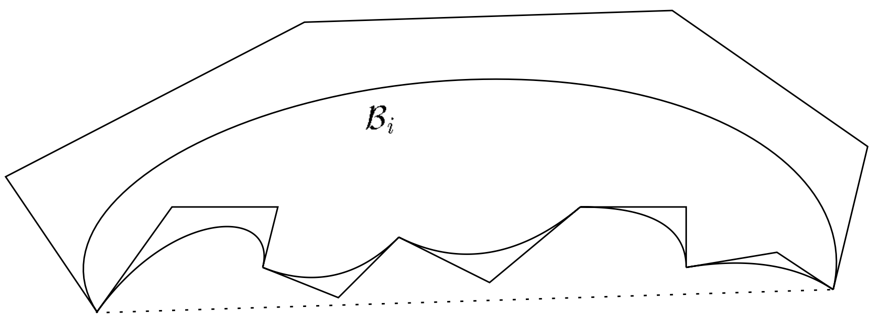

Figure III.6

III.4.2. *Présentation de l'algorithme*

Pour commencer cette sous-section, remarquons qu'en fait nous ne construisons pas une approximation polygonale à ε-près de l'enveloppe convexe d'une polycourbe, mais un objet qui approche cette enveloppe convexe à ε-près et a l'inconvénient de ne pas être un polygone et l'avantage d'avoir une construction beaucoup moins coûteuse qu'un polygone satisfaisant cette propriété. En effet, à chaque fois qu'une courbe de Bézier correspond aux conditions du lemme III.3, nous arrêtons de la subdiviser, même si la distance de Hausdorff entre elle et son B-polygone est supérieure à la valeur ε donnée, et au moment de l'écriture des résultats de l'algorithme nous remplaçons tout simplement le B-polygone par la courbe. Ceci fait que dans le cas d'une polycourbe convexe, par exemple, aucune subdivision ne sera nécessaire et l'algorithme nous rendra comme frontière de l'enveloppe convexe de la polycourbe la polycourbe elle-même. Dans le cas général (d'une polycourbe non-convexe), le résultat de l'algorithme sera une liste d'objets qui forment, en ordre trigonométrique, la frontière de l'approximation de l'enveloppe convexe de la polycourbe. Ces objets peuvent être soit des segments, soit des `courbes_de_Bézier`. Les segments pouvant eux aussi être écrits comme des `courbes_de_Bézier`, le résultat sera en fait une `polycourbe`.

Voici quelle est la structure de l'algorithme que nous proposons pour le calcul d'une approximation à ε près, pour un ε donné, de l'enveloppe convexe d'une polycourbe :

1. `Enveloppe_convexe_approchée(polycourbe` $\mathcal{B}$, `réel` ε, `polycourbe` $\mathcal{AE}$)
2. `tableau_pointeurs I`
3. `liste_courbes` $\mathcal{L}$
4. `begin`
5. `lecture_données(`$\mathcal{B}$,ε)
6. `enveloppe_convexe_polygone_contrôle(`$\mathcal{B}$,$\mathcal{L}$)
7. `dist_inf_eps(`$\mathcal{B}$,$\mathcal{L}$,ε,`I)`
8. `tant que I`$\neq \emptyset$
9.   `begin`
10.   `subdivision(`$\mathcal{B}$,`I)`
11.   `enveloppe_convexe_polygone_contrôle(`$\mathcal{B}$,$\mathcal{L}$)
12.   `dist_inf_eps(`$\mathcal{B}$,$\mathcal{L}$,ε,`I)`
13.   `end`
14. `écriture_resultats(`$\mathcal{B}$,$\mathcal{AE}$)
15. `end`

La seule procédure appelée par ce programme qui n'a pas encore été présentée est `dist_inf_eps` (lignes 7, 12). Nous la détaillons donc avant de commenter l'algorithme `Enveloppe_convexe_approchée`.

III.4.2.1. La procédure `dist_inf_eps`

Cette procédure prend comme paramètre la `polycourbe` $\mathcal{B}$, la `liste_courbes` $\mathcal{L}$ calculée par la procédure `enveloppe_convexe_polygone_contrôle` et le `réel` ε et retourne le `tableau_pointeurs J` qui identifie les courbes de Bézier de la composition de la polycourbe qui doivent être subdivisées. Une courbe de Bézier $\mathcal{B}_i$ de la structure de la polycourbe doit être subdivisée si, premièrement, elle se retrouve dans $\mathcal{L}$ avec le caractère associé "y" (voir le paragraphe III.3.2.2 pour précisions), et deuxièmement, nous ne pouvons pas affirmer que la distance de Hausdorff entre elle et son B-polygone, $\delta^{\mathrm{H}}(\mathcal{B}_i, \mathcal{P}_i)$, est inférieure à ε. Voilà maintenant le schéma de cet algorithme :

`Dist_inf_eps(polycourbe` $\mathcal{B}$, `liste_courbes` $\mathcal{L}$, `réel` ε, `tableau_pointeurs I)`
`entiers k, i`
`begin`
`k = nombre d'éléments de` $\mathcal{L}$
`pour i allant de 0 à k-1`
    `si` $\mathcal{L}$`[i].état_courbe = "y"` *et* $\delta^{\mathrm{H}}(\mathcal{P}_{\mathrm{j}}, [\mathrm{P}_0^{(\mathrm{j})}\mathrm{P}_{m_{\mathrm{j}}}^{(\mathrm{j})}]) \geq \varepsilon$
        `ajouter` $\mathcal{L}$`[i].ident_courbe à I`
`end`

La deuxième condition du test de la ligne 6 est celle donnée par le lemme III.1. Mais, en utilisant les notations du lemme, il est facile à voir que

$$\delta^{\mathrm{H}}(\mathcal{P}_0, [\mathrm{P}_0\mathrm{P}_m]) = \max_{i\in\{0,\ldots,m\}} \mathrm{d}(\mathrm{P}_i, [\mathrm{P}_0\mathrm{P}_m])$$

En effet, nous avons

$$\mathrm{d}(\mathrm{P}, [\mathrm{P}_0\mathrm{P}_m]) \leq \max\{\mathrm{d}(\mathrm{P}_i, [\mathrm{P}_0\mathrm{P}_m]), \mathrm{d}(\mathrm{P}_{i+1}, [\mathrm{P}_0\mathrm{P}_m])\} \qquad \forall \mathrm{P} \in [\mathrm{P}_i\mathrm{P}_{i+1}]$$

et nous obtenons tout de suite $\delta^{\mathrm{H}}(\mathcal{P}_0, [\mathrm{P}_0\mathrm{P}_m]) \leq \max_{i\in\{0,\ldots,m\}} \mathrm{d}(\mathrm{P}_i, [\mathrm{P}_0\mathrm{P}_m])$. Pour l'inégalité inverse, si nous supposons $\delta^{\mathrm{H}}(\mathcal{P}_0, [\mathrm{P}_0\mathrm{P}_m]) < \max_{i\in\{0,\ldots,m\}} \mathrm{d}(\mathrm{P}_i, [\mathrm{P}_0\mathrm{P}_m])$ nous déduisons immédiatement que le (ou les) sommet qui réalise(nt) $\max_{i\in\{0,\ldots,m\}} \mathrm{d}(\mathrm{P}_i, [\mathrm{P}_0\mathrm{P}_m])$ ne peu(ven)t pas appartenir à $\mathrm{B}([\mathrm{P}_0\mathrm{P}_m], \delta^{\mathrm{H}}(\mathcal{P}_0, [\mathrm{P}_0\mathrm{P}_m])$, d'où une contradiction. L'égalité ci-dessus est donc vérifiée. Notre algorithme va alors tester, pour toutes les courbes identifiées par la `liste_courbes` $\mathcal{L}$ comme étant "subdivisables", donc celle pour lesquelles le champ `état_courbe` est égal à "y", si le maximum des distances des sommets du polygone de contrôle au segment $\mathrm{P}_0\mathrm{P}_m$ joignant les extrémités de la courbe est inférieur à ε, et si ce n'est pas le cas un pointeur à la courbe est ajouté à la liste `I`. La complexité de ce test est linéaire en le nombre de sommets du B-polygone à tester.

Pour trouver la complexité du cycle des lignes 5-7, il faut sommer pour `i` allant de 0 à `k-1` les complexités des pas. En notant ℓ_i le nombre de sommets du B-polygone donné par *($\mathcal{L}$[`i`]`.ident_courbe`), le pas `i` est linéaire en ℓ_i, car le premier teste de la condition `si` et l'instruction à exécuter si les deux testes de cette condition sont satisfaits sont de complexité $\mathcal{O}(1)$ et le deuxième teste de la condition `si` est de complexité $\mathcal{O}(\ell_i)$. La complexité du cycle sera alors $\sum_{i=0}^{\mathrm{k}-1} \ell_i$, somme qui est inférieure ou égale à $M + n$, où n est le nombre de courbes de Bézier composant la polycourbe (l'égalité est atteinte si toutes les courbes de la composition de la polycourbe sont identifiées par la `liste_courbes` $\mathcal{L}$, car les sommets du polygone $\mathcal{P}$ qui sont des extrémités de courbe de Bézier de la composition de la polycourbe devront être comptés deux fois). La procédure `dist_inf_eps` sera donc linéaire en M.

III.4.2.2. Commentaire de l'algorithme `Enveloppe_convexe_approchée`

Toutes les procédures auxquelles cet algorithme fait appel étant présentées en détail, le fonctionnement du programme est facile à suivre. Le premier pas est le calcul de l'enveloppe convexe du polygone de contrôle de la polycourbe et l'obtention de la liste $\mathcal{L}$ des courbes de Bézier susceptibles de devoir être subdivisées ; ceci est fait par l'appel de `enveloppe_convexe_polygone_contrôle` de la ligne 6. Ensuite, la procédure `dist_inf_eps` établit lesquelles des courbes de Bézier apparaissant dans la liste $\mathcal{L}$ doivent vraiment être subdivisées, en vérifiant pour chacune si la majoration (donnée par le lemme III.1) de la distance de Hausdorff entre la courbe et son B-polygone est inférieure à ε. Des pointeurs identifiant toutes les courbes de Bézier qui devront être subdivisées se trouveront ainsi dans l'ensemble `I`. Évidemment, si cet ensemble est vide, il n'y a aucune subdivision à faire et nous finissons en écrivant le résultat obtenu. Si `I` est non-vide, les courbes qu'il désignes sont subdivisées et la composition de la polycourbe $\mathcal{B}$ est mise à jour par la procédure `subdivision` ; après, nous recommençons ces mêmes opérations jusqu'à l'obtention d'un ensemble I vide. La convergence de cet algorithme est assurée par la convergence du B-polygone vers la courbe de Bézier qui lui est associée par la subdivision de de Casteljau.

III.4.3. *La complexité de l'algorithme* `Enveloppe_convexe_approchée`

Commençons par étudier l'évolution de la taille de la polycourbe (le nombre de courbes de Bézier la composant) au fur à mesure qu'elle passe par les itérations du cycle des lignes 8-13.

Supposons que la polycourbe initiale s'écrit $\mathcal{B} = \cup_{i=1}^{n_0} \mathcal{B}_i$. En considérant qu'après k itérations la courbe $\mathcal{B}_i$ est décomposée en l'union des courbes de Bézier $\mathcal{B}_i^1, \mathcal{B}_i^2, \ldots, \mathcal{B}_i^{k_0}$, il est évident que si deux des points de contrôle de cette union de courbes sont des sommets de $\mathcal{Q}$, alors tous les points de contrôle qui sont "entre eux" seront des sommets de $\mathcal{Q}$. Autrement dit, si nous notons

$$\widetilde{\mathcal{P}}_i = \bigcup_{j=0}^{k_0} \mathcal{P}_i^j = \widetilde{\mathrm{P}}_0^{(i)} \widetilde{\mathrm{P}}_1^{(i)} \ldots \widetilde{\mathrm{P}}_{k_0 m_i}^{(i)}$$

alors soit tous les sommets de $\widetilde{\mathcal{P}}_i$ sont des sommets de $\mathcal{Q}$, cas auquel il n'y a aucune des courbes $\mathcal{B}_i^1, \ldots, \mathcal{B}_i^{k_0}$ à subdiviser, soit aucun des sommets de $\widetilde{\mathcal{P}}_i$ n'est un sommet de $\mathcal{Q}$, cas auquel il n'y a toujours aucune courbe à subdiviser, soit il existe $\ell_1, \ell_2 \in \{0, 1, \ldots, k_0 m_i\}$ tel que $\widetilde{\mathrm{P}}_0^{(i)}, \widetilde{\mathrm{P}}_1^{(i)}, \ldots, \widetilde{\mathrm{P}}_{\ell_1 - 1}^{(i)}$ et $\widetilde{\mathrm{P}}_{\ell_2+1}^{(i)}, \widetilde{\mathrm{P}}_{\ell_2+2}^{(i)}, \ldots, \widetilde{\mathrm{P}}_{k_0 m_i}^{(i)}$ ne soient pas des sommets de $\mathcal{Q}$ et $\widetilde{\mathrm{P}}_{\ell_1}^{(i)}, \widetilde{\mathrm{P}}_{\ell_1+1}^{(i)}, \ldots, \widetilde{\mathrm{P}}_{\ell_2}^{(i)}$ le soient, où ℓ_1 peut être égal à 0 et ℓ_2 peut être égal à $k_0 m_i$, mais pas les deux en même temps. Cette propriété est due au fait que la courbe de Bézier $\mathcal{B}_i$ est complètement convexe, ce qui impliquera la convexité du polygone $\widetilde{\mathcal{P}}_i$. Alors les seules courbes qui peuvent avoir à être subdivisées sont celles dont les points $\widetilde{\mathrm{P}}_{\ell_1}^{(i)}$ et $\widetilde{\mathrm{P}}_{\ell_2}^{(i)}$ sont, respectivement, des points de contrôle, donc il y aura au plus deux courbes à subdiviser (il peut toutefois n'y avoir qu'une seule ou même aucune dans ce cas aussi). Donc pour chacune des courbes de la composition initiale de la polycourbe il y a au plus deux courbes à subdiviser à chaque itération.

En conclusion, à chaque itération il y a au plus $2n_0$ courbes de Bézier à subdiviser, et par conséquent après k itérations la polycourbe sera composée d'au plus $2kn_0$ courbes de Bézier.

Pour calculer la complexité de `Enveloppe_convexe_approchée`, nous devons d'abord trouver une estimation pour le nombre d'itérations nécessaires pour que la procédure `dist_inf_eps` construise un ensemble I vide, car c'est celle-là la condition de sortie du cycle `tant que` des lignes 8-13. Ce qui nous intéresse est donc le nombre de subdivisions qui doivent être appliquées à une courbe de Bézier donnée $\mathcal{B}_i$ pour que la deuxième condition du test de la ligne 8 de la procédure `dist_inf_eps` soit satisfaite par les (au plus deux) courbes de Bézier $\mathcal{B}_i^{k_1}$ et $\mathcal{B}_i^{k_2}$ (de la composition de $\mathcal{B}_i$ après subdivision) dont le champ associé dans la `liste_courbes` $\mathcal{L}$ n'a pas la valeur "n". En notant $\mathcal{B}_i(k)$ une des courbes obtenues de $\mathcal{B}_i$ après k subdivisions, nous savons que $\delta^{\mathrm{H}}(\mathcal{P}_i(k), [\mathrm{P}_0^{(i)}(k)\mathrm{P}_{m_i}^{(i)}(k)])$ converge vers 0 quand k tend vers l'infini et nous voulons estimer sa vitesse de convergence.

Si $\mathcal{B}_0$ est une courbe de Bézier et $\mathcal{B}_0'$ et $\mathcal{B}_0''$ sont les deux courbes de Bézier obtenues de $\mathcal{B}_0$ par l'application une fois de la subdivision de de Casteljau de rapport 1/2, alors nous avons l'inégalité [Fio87]

$$\max\{\max_{i\in\{0,\ldots,m-1\}} \mathrm{d}(\mathrm{P}'_i,\mathrm{P}'_{i+1}), \max_{i\in\{0,\ldots,m-1\}} \mathrm{d}(\mathrm{P}''_i,\mathrm{P}''_{i+1})\} \leq \frac{1}{2}\max_{i\in\{0,\ldots,m-1\}} \mathrm{d}(\mathrm{P}_i,\mathrm{P}_{i+1})$$

Nous obtenons par récurrence

$$\max_{j\in\{1,\ldots,2^k\}}\max_{i\in\{0,\ldots,m-1\}} \mathrm{d}(\mathrm{P}^j_i,\mathrm{P}^j_{i+1}) \leq \frac{1}{2^k}\max_{i\in\{0,\ldots,m-1\}} \mathrm{d}(\mathrm{P}_i,\mathrm{P}_{i+1}) \quad \forall k\in \mathbb{N},$$

où $\mathrm{P}^j_0\mathrm{P}^j_1\ldots\mathrm{P}^j_m$ est le polygone de contrôle de la j-ème courbe de la structure de $\mathcal{B}_0$ après k subdivisions.

Malheureusement, l'inégalité similaire pour les distances des points de contrôle au segment $[\mathrm{P}^j_0\mathrm{P}^j_m]$ n'est pas vérifiée, car nous pouvons prouver par un calcul simple que pour le cas de la figure III.7 nous avons la relation

$$2\mathrm{d}(\mathrm{P}'_2,[\mathrm{P}_0\mathrm{P}'_5]) = 2\mathrm{d}(\mathrm{P}'_2,\mathrm{P}_0) > \mathrm{d}(\mathrm{P}_2,\mathrm{P}_0) = \mathrm{d}(\mathrm{P}_3,\mathrm{P}_5) = \max_{0\leq i\leq 5}\mathrm{d}(\mathrm{P}_i,[\mathrm{P}_0\mathrm{P}_5])$$

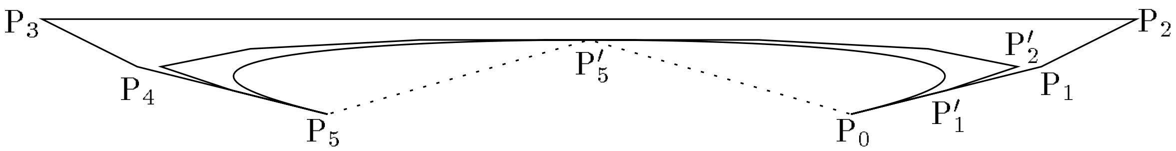

Figure III.7

Par conséquent, nous allons utiliser la majoration

$$\max_{i\in\{0,\ldots,m\}} \mathrm{d}(\mathrm{P}^j_i,[\mathrm{P}^j_0\mathrm{P}^j_m]) < \frac{1}{2}\sum_{i=0}^{m-1}\mathrm{d}(\mathrm{P}^j_i,\mathrm{P}^j_{i+1}),$$

qui est évidemment vérifiée et qui nous permet d'écrire

$$\max_{i\in\{0,\ldots,m\}} \mathrm{d}(\mathrm{P}^j_i,[\mathrm{P}^j_0\mathrm{P}^j_m]) < \frac{1}{2^k}\sum_{i=0}^{m-1}\mathrm{d}(\mathrm{P}_i,\mathrm{P}_{i+1})$$

ou encore $\max_{i\in\{0,\ldots,m\}} \mathrm{d}(\mathrm{P}^j_i,[\mathrm{P}^j_0\mathrm{P}^j_m]) < \frac{1}{2^k}S$, où S est une constante qui dépend de la courbe de Bézier.

Supposons que la condition qui nous intéresse est satisfaite après k_0 itération. Nous aurons alors la suite d'implications

$$\frac{1}{2^{k_0}}S < \varepsilon \Longrightarrow 2^{k_0} > \frac{S}{\varepsilon} \Longrightarrow k_0 > \log_2\frac{S}{\varepsilon}$$

ce qui nous donne le nombre d'itérations nécessaires pour que la `condition_arrêt` soit satisfaite comme étant le plus petit entier supérieur à $\log_2(S/\varepsilon)$.

Soit d le maximum des degrés des n_0 courbes de Bézier composant la polycourbe initiale. Son polygone de contrôle a au plus n_0d sommets. Nous avons vu qu'après k itérations la polycourbe sera composée d'au plus $2kn_0$ courbes de Bézier, pour $k>0$. Ainsi, le polygone de contrôle de la polycourbe aura, après k itérations, au plus $2n_0dk$ sommets.

Pour calculer la complexité de l'algorithme `enveloppe_convexe_approchée`, il faut sommer les coûts des procédures appelées par cet algorithme. La procédure de la ligne 6 est de complexité $\mathcal{O}(n_0d)$ et celle de la ligne 7 aussi. Pour calculer le coût du cycle des lignes 8-13, il faut sommer les coûts de la k-ième itération pour k allant de 1 à

$[\log_2(S/\varepsilon)] + 1 = \lceil \log_2(S/\varepsilon) \rceil$, où $[x]$ désigne la partie entière de x. À la k-ième itération, la procédure de la ligne 10 coûtera $\mathcal{O}(n_0 d^2 k)$, celle de la ligne 11 coutera $\mathcal{O}(n_0 dk)$ et celle de la ligne 12 coutera aussi $\mathcal{O}(n_0 dk)$. Nous obtenons donc la complexité de la k-ième itération du cycle égale à $\mathcal{O}(n_0 d^2 k)$. En sommant, la complexité du cycle entier sera

$$\mathcal{O}\big(n_0 d^2 \lceil \log_2(S/\varepsilon) \rceil (\lceil \log_2(S/\varepsilon) \rceil + 1)\big)$$

ou, plus grossiérement, $\mathcal{O}\big(n_0 d^2 \lceil \log_2(S/\varepsilon) \rceil^2\big)$. Évidemment, le coût de l'algorithme sera, au total, le même, et nous rappelons que S est une constante, qui dépend de la polycourbe.

III.5. L'enveloppe convexe topologique

Dans le chapitre précédent, nous avons vu que l'enveloppe convexe du polygone de contrôle d'une polycourbe de Bézier sans tangence intérieure converge topologiquement vers l'enveloppe convexe de la polycourbe si on applique au polygone de contrôle un nombre suffisant de subdivisions. Nous donnerons dans ce paragraphe un algorithme qui calcule pour une polycourbe donnée par son polygone de contrôle initial un polygone de contrôle tel que son enveloppe convexe soit topologiquement pertinente pour celle de la polycourbe.

La condition de pertinence topologique donnée par la preuve du théorème II.9 est suffisante, mais pas nécessaire, et elle est évidemment inutilisable pour notre algorithme, faisant intervenir des distances qui ne sont pas calculables uniquement à partir du polygone de contrôle de la polycourbe. L'idéal serait bien sûr de trouver une condition nécessaire et suffisante, une telle condition minimisant le nombre de subdivisions à faire, mais la notion de pertinence topologique dont nous avons besoin est par trop heuristique et nous avons seulement trouvé des conditions suffisantes pour notre problème.

III.5.1. *Résultats théoriques*

En utilisant les notations de l'énoncé du lemme III.2, soient $\mathcal{K} = \{1, \ldots, p\}$ et $\mathcal{K}_\alpha = \{k \in \mathcal{K} \mid i_k \in \mathcal{I}_\alpha\}$ pour $\alpha = 0, 1, 2$. Pour éviter les cas similaires au celui de la figure III.4, nous supposerons card $\mathcal{K} > 1$.

III.5.1.1. Conditions suffisantes pour la pertinence topologique

Proposition III.4. – *Si les trois conditions suivantes sont satisfaites, alors l'enveloppe convexe du polygone de contrôle, $\mathcal{E}(\mathcal{P})$, est topologiquement pertinente pour l'enveloppe convexe de la polycourbe, $\mathcal{E}(\mathcal{B})$:*

(III.4.i) *si* P *est un sommet de $\mathcal{P}$ tel que* $\mathrm{P} \notin \mathrm{DP}\big(\mathrm{P}_0^{(i_k)}\mathrm{P}_1^{(i_k)}, \mathcal{B}_{i_k}\big)$, *alors* P *est un sommet de $\mathcal{P}_{i_{k-1}}$;*

(III.4.ii) *si* P *est un sommet de $\mathcal{P}$ tel que* $\mathrm{P} \notin \mathrm{DP}\big(\mathrm{P}_{m_{i_k}-1}^{(i_k)}\mathrm{P}_{m_{i_k}}^{(i_k)}, \mathcal{B}_{i_k}\big)$, *alors* P *est un sommet de $\mathcal{P}_{i_{k+1}}$;*

(III.4.iii) *si* $\mathcal{B}_{i_{k-1}}.\texttt{rang} \neq \mathcal{B}_{i_k}.\texttt{rang}$ *et* $\mathcal{B}_{i_{k+1}}.\texttt{rang} \neq \mathcal{B}_{i_k}.\texttt{rang}$, *alors*

$$\{\mathrm{P}_0^{(i_k)}, \mathrm{P}_{m_{i_k}}^{(i_k)}\} \not\subset \mathrm{DP}\big(\mathrm{PQ}, \mathcal{B}_{i_{k-1}}\big),$$

où P *est un sommet de $\mathcal{P}_{i_{k-1}}$ et* Q *un sommet de $\mathcal{P}_{i_{k+1}}$ tels que le segment* [PQ] *soit une arête du polygone frontière de* $\mathcal{E}\big(\cup_{\alpha\neq k}\mathcal{P}_{i_\alpha}\big)$.

Les deux figures suivantes illustrent les trois conditions données par cette proposition.

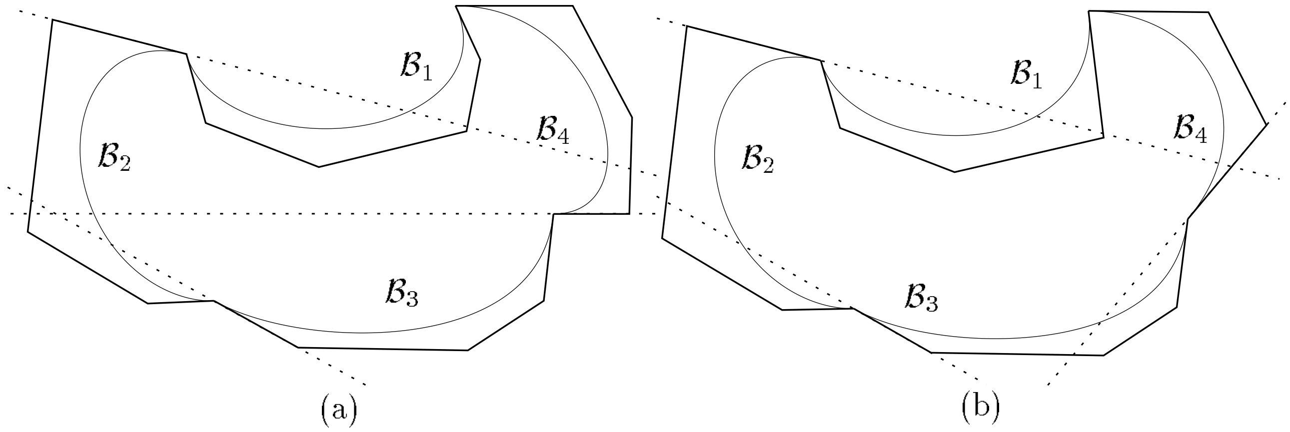

Figure III.8 (a) *les courbes* $\mathcal{B}_2$ *et* $\mathcal{B}_4$ *ne vérifient pas la condition* (III.4.i) *;* (b) *la condition* (III.4.i) *est vérifiée* $\forall k \in \mathcal{K}$. *La propriété* (III.4.ii) *est symétrique avec* (III.4.i).

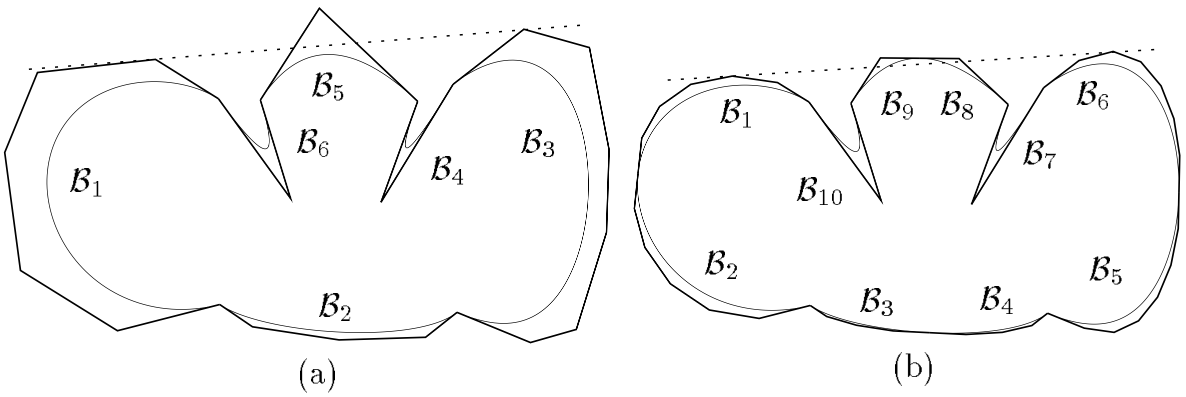

Figure III.9 (a) *la courbe* $\mathcal{B}_5$ *ne satisfait pas la condition* (III.4.iii) *;* (b) *après une itération, les courbes* $\mathcal{B}_8$ *et* $\mathcal{B}_9$ *obtenues par subdivision de la courbe mentionnée avant satisfont toutes les deux cette condition.*

Preuve de la proposition – Nous allons démontrer que si les trois conditions ci-dessus sont satisfaites, alors les deux propriétés de la définition II.6 de la pertinence topologique sont vérifiées par $\mathcal{P}$ et $\mathcal{B}$. Nous commençons par prouver que (II.6.i) est satisfaite.

Soit $i \in \{1, \ldots, n\}$ tel que $\mathcal{P}_i \cap \mathcal{Q} \neq \emptyset$. Alors il existe $k \in \mathcal{K}$ tel que $i = i_k$. Si $k \in \mathcal{K}_1 \cup \mathcal{K}_2$, alors $\mathcal{P}_{i_k} \cap \mathcal{Q} \subset \big\{\mathrm{P}_0^{(i_k)}, \mathrm{P}_{m_{i_k}}^{(i_k)}\big\} \subset \mathcal{B}_{i_k}$ et, comme $\mathcal{E}(\mathcal{B}) \subset \mathcal{E}(\mathcal{P})$, le fait que $\mathcal{B}_{i_k}$ et $\mathcal{Q}$ ne sont pas disjoints impliquera $\mathcal{B}_{i_k} \cap \mathcal{E} \neq \emptyset$.

Si $k \in \mathcal{K}_0$, supposons que $\mathcal{B}_{i_{k-1}}.\texttt{rang} = \mathcal{B}_{i_k}.\texttt{rang}$. Le polygone

$$\widetilde{\mathcal{P}}_{i_k} = \bigcup_{\mathcal{B}_i.\mathrm{rang} = \mathcal{B}_{i_k}.\mathrm{rang}} \mathcal{P}_i$$

étant convexe, nous aurons forcément $i_{k-1} = i_k - 1$, car, comme nous avons déjà vu dans le paragraphe III.4.3, si deux sommets de $\widetilde{\mathcal{P}}_{i_k}$ sont des sommets de $\mathcal{Q}$, alors tous les sommets de $\widetilde{\mathcal{P}}_{i_k}$ situés "entre" les deux seront des sommets de $\mathcal{Q}$. Par conséquent, $\mathrm{P}_0^{(i_k)}$ est un sommet de $\mathcal{Q}$ et nous sommes de nouveau dans la situation $\mathcal{B}_{i_k} \cap \mathcal{Q} \neq \emptyset$. Un raisonnement tout à fait identique nous mène à la même conclusion si nous supposons $\mathcal{B}_{i_{k+1}}.\mathtt{rang} = \mathcal{B}_{i_k}.\mathtt{rang}$.

Supposons maintenant que $\mathcal{B}_{i_{k-1}}.\mathtt{rang} \neq \mathcal{B}_{i_k}.\mathtt{rang}$ et $\mathcal{B}_{i_{k+1}}.\mathtt{rang} \neq \mathcal{B}_{i_k}.\mathtt{rang}$. Nous allons prouver que $\mathcal{B}_{i_k} \cap \mathcal{E} \neq \emptyset$ par l'absurde, en supposant $\mathcal{B}_{i_k} \subset \mathrm{Int}\big(\mathcal{E}(\cup_{i\neq i_k}\mathcal{B}_i)\big)$. Comme $\{\mathrm{P}_0^{(i_k)}, \mathrm{P}_{m_{i_k}}^{(i_k)}\} \not\subset \mathrm{DP}(\mathrm{PQ}, \mathcal{B}_{i_k})$, avec P et Q définis comme dans la condition (III.4.iii), nous aurons $\{\mathrm{P}_0^{(i_k)}, \mathrm{P}_{m_{i_k}}^{(i_k)}\} \not\subset \mathcal{E}(\cup_{\alpha\neq k}\mathcal{P}_{i_\alpha})$, donc $\mathcal{B}_{i_k} \not\subset \mathcal{E}(\cup_{\alpha\neq k}\mathcal{P}_{i_\alpha})$. Mais $\mathcal{B}_{i_k} \subset \mathrm{Int}\big(\mathcal{E}(\cup_{i\neq i_k}\mathcal{B}_i)\big)$, par conséquent il existe $i_0 \in \{1, \ldots, n\}$ tel que

$$\mathcal{B}_{i_0} \not\subset \mathcal{E}\Big(\bigcup_{\alpha\neq k} \mathcal{B}_{i_\alpha} \cup \mathcal{B}_{i_k}\Big) = \mathcal{E}\Big(\bigcup_{\alpha\in\mathcal{K}} \mathcal{B}_{i_\alpha}\Big)$$

D'un autre côté, si nous notons

$$\mathcal{R}_{i_k} = \mathrm{DP}(\mathrm{P}_0^{(i_k)}\mathrm{P}_1^{(i_k)}, \mathcal{B}_{i_k}) \cap \mathrm{DP}(\mathrm{P}_{m_{i_k}-1}^{(i_k)}\mathrm{P}_{m_{i_k}}^{(i_k)}, \mathcal{B}_{i_k}),$$

nous aurons (conformément aux conditions (III.4.i) et (III.4.iii)), $\cup_{i\notin\{i_{k-1},i_{k+1}\}}\mathcal{P}_i \subset \mathcal{R}_{i_k}$ et aussi, évidemment, $\cup_{i\notin\{i_{k-1},i_{k+1}\}}\mathcal{B}_i \subset \mathcal{R}_{i_k}$.

L'ensemble $\mathcal{R}_{i_k} \setminus \mathcal{B}_{i_k}$ a deux composantes connexes,

$$\mathcal{R}'_{i_k} = \bigcap_{t\in[0,1]} \mathrm{DP}(\Delta_t, \mathcal{B}_{i_k}) \quad \text{et} \quad \mathcal{R}''_{i_k} = \mathcal{R}_{i_k} \setminus \big(\mathcal{R}'_{i_k} \cup \mathcal{B}_{i_k}\big),$$

où Δ_t est la droite tangente à la courbe $\mathcal{B}_{i_k}$ en le point $\mathrm{B}_{m_{i_k}}\big(\mathrm{P}_0^{(i_k)}, \mathrm{P}_1^{(i_k)}, \ldots, \mathrm{P}_{m_{i_k}}^{(i_k)}\,;\, t\big)$.

Nous aurons soit $\cup_{i\notin\{i_{k-1},i_k,i_{k+1}\}}\mathcal{B}_i \subset \mathcal{R}'_{i_k}$, soit $\cup_{i\notin\{i_{k-1},i_k,i_{k+1}\}}\mathcal{B}_i \subset \mathcal{R}''_{i_k}$, car $\mathcal{B}$ est une courbe simple et donc $\cup_{i\notin\{i_{k-1},i_k,i_{k+1}\}}\mathcal{B}_i$, qui est une courbe continue, ne peut pas "traverser" $\mathcal{B}_{i_k}$, la séparation entre les deux régions $\mathcal{R}'_{i_k}$ et $\mathcal{R}''_{i_k}$.

En supposant $\cup_{i\notin\{i_{k-1},i_k,i_{k+1}\}}\mathcal{B}_i \subset \mathcal{R}'_{i_k}$, nous aurons $\mathcal{E}\big(\cup_{i\notin\{i_{k-1},i_k,i_{k+1}\}}\mathcal{B}_i\big) \subset \mathcal{R}'_{i_k}$, car $\mathcal{R}'_{i_k}$ est convexe et, comme $\mathcal{B}_{i_k} \not\subset \mathcal{E}\big(\cup_{\alpha\neq k}\mathcal{B}_{i_\alpha}\big)$, nous obtenons tout de suite une contradiction avec l'hypothèse $\mathcal{B}_{i_k} \subset \mathrm{Int}\big(\mathcal{E}(\cup_{i\neq i_k}\mathcal{B}_i)\big)$. Il nous reste ainsi $\cup_{i\notin\{i_{k-1},i_k,i_{k+1}\}}\mathcal{B}_i \subset \mathcal{R}''_{i_k}$. Cette région est encore séparée en deux parties connexes par la ligne polygonale $\mathcal{P}_{i_k}$, les deux parties étant $reg(\mathcal{P}_{i_k}, \mathcal{B}_{i_k})$ et $\mathcal{R}'''_{i_k} = \mathcal{R}''_{i_k} \setminus \big(reg(\mathcal{P}_{i_k}, \mathcal{B}_{i_k}) \cup \mathcal{P}_{i_k}\big)$. Le polygone de contrôle de la polycourbe $\mathcal{P}$ est pseudo-simple et card $\mathcal{K} > 1$, ce qui implique $\mathcal{P} \not\subset \mathcal{E}(\mathcal{P}_{i_k})$, d'où il résulte $\mathcal{P} \cap reg(\mathcal{P}_{i_k}, \mathcal{B}_{i_k}) = \emptyset$. Alors $\cup_{i\notin\{i_{k-1},i_k,i_{k+1}\}}\mathcal{B}_i \subset \mathcal{R}'''_{i_k}$ et comme le polygone $\mathcal{P}$ est une courbe fermée nous obtenons immédiatement

$$\mathcal{P}_{i_k} \setminus \{\mathrm{P}_0^{(i_k)}, \mathrm{P}_{m_{i_k}}\} \subset \mathrm{Int}\Big(\mathcal{E}\big(\bigcup_{i\neq i_k} \mathcal{P}_{i_k}\big)\Big),$$

ce qui contredit l'hypothèse dans la quelle nous travaillons, $k \in \mathcal{K}_0$.

Ainsi, l'hypothèse $\mathcal{B}_{i_k} \subset \mathrm{Int}\big(\mathcal{E}(\cup_{i\neq i_k}\mathcal{B}_i)\big)$ nous a menés dans tous les cas à une contradiction, donc elle est fausse et la première propriété de la définition de la pertinence topologique est prouvée.

Pour la deuxième propriété de cette définition, (II.6.ii), nous allons faire aussi une preuve par l'absurde. Supposons qu'il existe $i \in \{1, \ldots, n\}$ tel que $\mathcal{P}_i \subset \mathrm{Int}\big(\mathcal{E}(\cup_{\alpha\neq i}\mathcal{P}_\alpha)\big)$ et $\mathcal{B}_i \not\subset \mathcal{E}\big(\cup_{\alpha\neq i}\mathcal{B}_\alpha\big)$. La condition $\mathcal{P}_i \subset \mathrm{Int}\big(\mathcal{E}(\cup_{\alpha\neq i}\mathcal{P}_\alpha)\big)$ implique $i \notin \mathcal{I} = \{i_k \mid k \in \mathcal{K}\}$. Des conditions $\mathcal{B}_i \not\subset \mathcal{E}\big(\cup_{\alpha\neq i}\mathcal{B}_\alpha\big)$ et $i \notin \mathcal{I} = \{i_k \mid k \in \mathcal{K}\}$ nous déduisons immédiatement $\mathcal{B}_i \not\subset \mathcal{E}\big(\cup_{k\in\mathcal{K}}\mathcal{B}_{i_k}\big)$. Nous aurons alors soit $\mathcal{B}_i \cap \mathcal{E}\big(\cup_{k\in\mathcal{K}}\mathcal{B}_{i_k}\big) = \emptyset$, soit $\mathcal{B}_i \cap \delta\big(\mathcal{E}(\cup_{k\in\mathcal{K}}\mathcal{B}_{i_k})\big) \neq$

$\emptyset$. Dans les deux cas, il doit exister $i_0 \in \{1, 2, \ldots, n\}$ tel que $\mathcal{B}_{i_0} \cap \delta(\mathcal{E}(\cup_{k\in\mathcal{K}}\mathcal{B}_{i_k})) \neq \emptyset$. Mais $\mathcal{B}$ est une courbe simple, donc il existe $k \in \mathcal{K}$ tel que $\mathcal{B}_{i_0} \cap]\mathrm{R}'_{i_k}\mathrm{R}_{i_{k+1}}[\neq \emptyset$, où $\mathrm{R}'_{i_k} \in \mathcal{B}_{i_k}$ et $\mathrm{R}_{i_{k+1}} \in \mathcal{B}_{i_{k+1}}$ sont deux points tels que la droite $\mathrm{R}'_{i_k}\mathrm{R}_{i_{k+1}}$ soit une tangente commune des courbes $\mathcal{B}_{i_k}$ et $\mathcal{B}_{i_{k+1}}$ et que $[\mathrm{R}'_{i_k}\mathrm{R}_{i_{k+1}}] \subset \delta(\mathcal{E}(\cup_{k\in\mathcal{K}}\mathcal{B}_{i_k}))$.

En utilisant les notations de la première partie de cette démonstration, les conditions (III.4.i) et (III.4.ii) impliquent $\mathcal{P}_{i_0} \subset \mathcal{R}_{i_k} \cap \mathcal{R}_{i_{k+1}}$ et, suite aux propriétés des courbes de Bézier, $\mathcal{B}_{i_0} \subset \mathcal{R}_{i_k} \cap \mathcal{R}_{i_{k+1}}$. Nous aurons alors $\mathcal{A} =]\mathrm{R}'_{i_k}\mathrm{R}_{i_{k+1}}[\cap (\mathcal{R}_{i_k} \cap \mathcal{R}_{i_{k+1}}) \neq \emptyset$. Comme $]\mathrm{R}'_{i_k}\mathrm{R}_{i_{k+1}}[\cap \mathcal{B}_{i_k} = \emptyset$ et $]\mathrm{R}'_{i_k}\mathrm{R}_{i_{k+1}}[\cap \mathcal{B}_{i_{k+1}} = \emptyset$, nous serons dans un des quatre cas suivants :

(i) $\mathcal{A} \subset \mathcal{R}'_{i_k} \cap \mathcal{R}'_{i_{k+1}}$	(ii) $\mathcal{A} \subset \mathcal{R}'_{i_k} \cap \mathcal{R}''_{i_{k+1}}$
(iii) $\mathcal{A} \subset \mathcal{R}''_{i_k} \cap \mathcal{R}'_{i_{k+1}}$	(iv) $\mathcal{A} \subset \mathcal{R}''_{i_k} \cap \mathcal{R}''_{i_{k+1}}$

Si $\mathcal{A} \subset \mathcal{R}'_{i_k}$, alors $\mathrm{R}'_{i_k} = \mathrm{P}^{(i_k)}_{m_{i_k}}$. En effet, soit $t_0 \in [0, 1]$ tel que

$$\mathrm{R}'_{i_k} = \mathrm{B}_{m_{i_k}}(\mathrm{P}^{(i_k)}_0, \ldots, \mathrm{P}^{(i_k)}_{m_{i_k}}; t_0)$$

Si $t_0 \in]0, 1[$, nous aurons évidemment $]\mathrm{R}'_{i_k}\mathrm{R}_{i_{k+1}}[\cap \mathcal{R}'_{i_k} = \emptyset$, car la courbe $\mathcal{B}_{i_k}$ est convexe. Aussi, si $t_0 = 0$, alors en posant la condition $]\mathrm{R}'_{i_k}\mathrm{R}_{i_{k+1}}[\cap \mathcal{R}'_{i_k} \neq \emptyset$ nous imposons un mauvais ordre entre les points R'_{i_k} et $\mathrm{R}_{i_{k+1}}$ sur la frontière de $\mathcal{E}(\cup_{k\in\mathcal{K}}\mathcal{B}_{i_k})$. Il reste donc $t_0 = 1$, avec la remarque que la droite $\mathrm{R}'_{i_k}\mathrm{R}_{i_{k+1}}$ ne doit pas être tangente à la courbe $\mathcal{B}_{i_k}$ en $\mathrm{P}_{m_{i_k}}$ pour avoir $]\mathrm{R}'_{i_k}\mathrm{R}_{i_{k+1}}[\cap \mathcal{R}'_{i_k} \neq \emptyset$.

De la même manière, si $\mathcal{A} \subset \mathcal{R}'_{i_{k+1}}$, alors $\mathrm{R}_{i_{k+1}} = \mathrm{P}^{(i_{k+1})}_0$. Les cas (i) à (iv) sont présentés figure III.10.

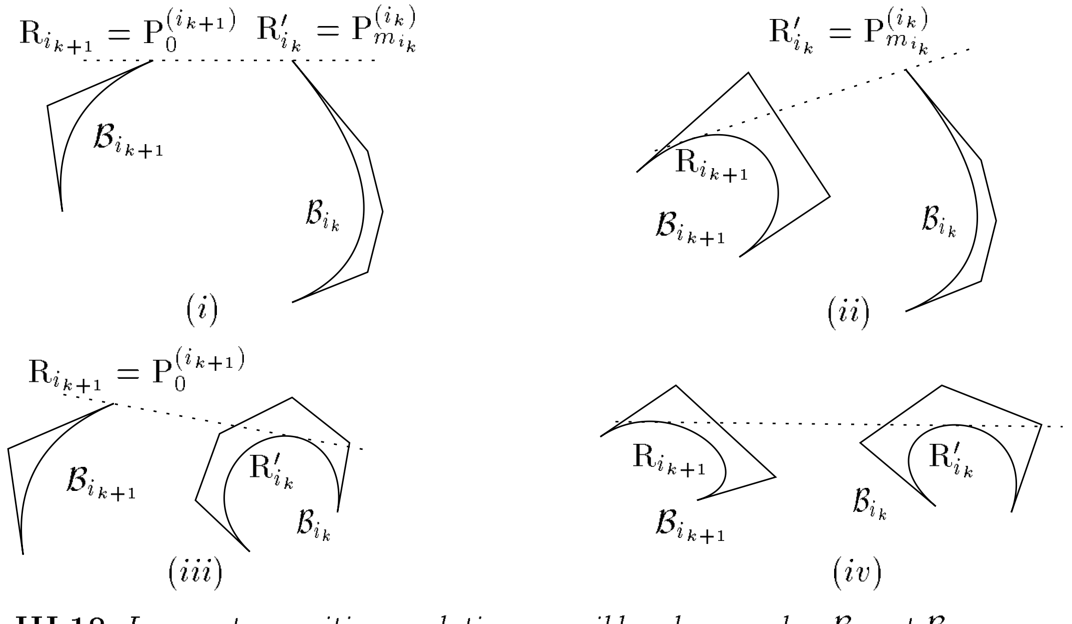

Figure III.10 *Les quatre positions relatives possibles des courbes $\mathcal{B}_{i_k}$ et $\mathcal{B}_{i_{k+1}}$*

Le cas (i) nous mène rapidement à une contradiction. En effet, $\mathcal{B}_{i_0} \cap]\mathrm{P}^{(i_k)}_{m_{i_k}}\mathrm{P}^{(i_{k+1})}_0[\neq \emptyset$ implique $\mathcal{P}_{i_0} \cap \mathrm{P}^{(i_k)}_{m_{i_k}}\mathrm{P}^{(i_{k+1})}_0 \neq \emptyset$, et comme $[\mathrm{P}^{(i_k)}_{m_{i_k}}\mathrm{P}^{(i_{k+1})}_0]$ est, dans ce cas, une arête du polygone $\mathcal{Q}$, nous avons obtenu une contradiction.

Les cas (ii) et (iii) étant symétriques, nous étudierons uniquement le cas (ii). Il y a deux situation possibles : soit $i_0 \in \{i_k + 1, \ldots, i_{k+1} - 1\}$, soit $i_0 \in \{i_{k+1} + 1, \ldots, i_k - 1\}$, l'ensemble d'indices $\mathcal{I}$ étant considéré circulaire. Supposons $i_0 \in \{i_k + 1, \ldots, i_{k+1} - 1\}$ et étudions la position possible de la ligne polygonale $\cup_{i=i_0+1}^{i_{k+1}-1}\mathcal{P}_i$. Cette ligne polygonale doit joindre les points $\mathrm{P}_{m_{i_0}}^{(i_0)}$ et $\mathrm{P}_0^{(i_{k+1})}$ et tous ses sommets, à part $\mathrm{P}_0^{(i_{k+1})}$, doivent appartenir à la région $\mathcal{R}_{i_{k+1}}$, car évidemment $i_{k+2} \notin \{i_0+1,\ldots,i_{k+1}-1\}$. Cette région est séparée en deux par la ligne polygonale $\mathrm{P}_1^{(i_{k+1})}\mathrm{P}_2^{(i_{k+1})}\ldots\mathrm{P}_{m_{i_{k+1}}-1}^{(i_{k+1})}$:

$$\widetilde{\mathcal{R}}_{i_{k+1}} = \bigcap_{j=0}^{m_{i_{k+1}}-1} \mathrm{DP}\big(\mathrm{P}_j^{(i_{k+1})}\mathrm{P}_{j+1}^{(i_{k+1})}, \mathcal{P}_{i_{k+1}}\big)$$

$$\overline{\mathcal{R}}_{i_{k+1}} = \mathcal{R}_{i_{k+1}} \setminus \Big(\widetilde{\mathcal{R}}_{i_{k+1}} \cup \bigcup_{j=1}^{m_{i_{k+1}}-2} [\mathrm{P}_j^{(i_{k+1})}\mathrm{P}_{j+1}^{(i_{k+1})}]\Big)$$

et, le polygone $\mathcal{P}$ étant pseudo-simple, les deux lignes polygonales, $\mathrm{P}_1^{(i_{k+1})}\ldots\mathrm{P}_{m_{i_{k+1}}-1}^{(i_{k+1})}$ et $\cup_{i=i_0+1}^{i_{k+1}-1}\mathcal{P}_i$, doivent être disjointes. Mais, d'un côté, $\mathrm{P}_{m_{i_0}}^{(i_{k+1})} \in \overline{\mathcal{R}}_{i_{k+1}}$, comme nous avons déjà vu pour un cas similaire dans la première partie de cette démonstration, et d'un autre côté il existe évidemment un voisinage $\mathcal{V}$ du point $\mathrm{P}_0^{(i_{k+1})} = \mathrm{P}_{m_{i_{k+1}}-1}^{(i_{k+1}-1)}$ tel que

$$\mathcal{V} \cap \widetilde{\mathcal{R}}_{i_{k+1}} \cap]\mathrm{P}_{m_{i_{k+1}}-1}^{(i_{k+1}-1)}\mathrm{P}_{m_{i_{k+1}}-2}^{(i_{k+1}-1)}] \neq \emptyset$$

Ainsi, nous avons

$$\bigcup_{i=i_0+1}^{i_{k+1}-1}\mathcal{P}_i \setminus \{\mathrm{P}_{m_{i_{k+1}}-1}^{(i_{k+1}-1)}\} \subset \mathcal{R}_{i_{k+1}}$$

$$\Big(\bigcup_{i=i_0+1}^{i_{k+1}-1}\mathcal{P}_i \setminus \{\mathrm{P}_{m_{i_{k+1}}-1}^{(i_{k+1}-1)}\}\Big) \cap \widetilde{\mathcal{R}}_{i_{k+1}} \neq \emptyset \qquad \Big(\bigcup_{i=i_0+1}^{i_{k+1}-1}\mathcal{P}_i \setminus \{\mathrm{P}_{m_{i_{k+1}}-1}^{(i_{k+1}-1)}\}\Big) \cap \overline{\mathcal{R}}_{i_{k+1}} \neq \emptyset$$

et par conséquent

$$\Big(\bigcup_{i=i_0+1}^{i_{k+1}-1}\mathcal{P}_i \setminus \{\mathrm{P}_{m_{i_{k+1}}-1}^{(i_{k+1}-1)}\}\Big) \cap \bigcup_{j=1}^{m_{i_{k+1}}-2} [\mathrm{P}_j^{(i_{k+1})}\mathrm{P}_{j+1}^{(i_{k+1})}] \neq \emptyset,$$

d'où la contradiction.

Supposons maintenant que $i_0 \in \{i_{k+1}+1,\ldots,i_k-1\}$. Soit $\mathrm{P}_0 \in \mathcal{B}_{i_0} \cap]\mathrm{R}'_{i_k}\mathrm{R}_{i_{k+1}}[$ tel que

$$\left(\bigcup_{i=i_{k+1}+1}^{i_0-1} \mathcal{B}_i \cup \mathcal{B}_{i_0}\big|_{]\mathrm{P}_0^{(i_0)}\mathrm{P}_0[}\right) \cap]\mathrm{R}'_{i_k}\mathrm{R}_{i_{k+1}}[= \emptyset,$$

où $\mathcal{B}_{i_0}\big|_{]\mathrm{P}_0^{(i_0)}\mathrm{P}_0[}$ est la partie de la courbe $\mathcal{B}_{i_0}$ comprise entre les points $\mathrm{P}_0^{(i_0)}$ et P_0, sans ces deux-là. Soit

$$\mathcal{L} = \left(\bigcup_{i=i_{k+1}+1}^{i_0-1} \mathcal{B}_i \cup \mathcal{B}_{i_0}\big|_{]\mathrm{P}_0^{(i_0)}\mathrm{P}_0[}\right) \cup [\mathrm{P}_0\mathrm{R}_{i_{k+1}}] \cup \mathcal{B}_{i_{k+1}}\big|_{]\mathrm{R}_{i_{k+1}}\mathrm{P}_{m_{i_{k+1}}}^{(i_{k+1})}[}$$

De par sa définition, la courbe $\mathcal{L}$ est simple et fermée. En plus, nous pouvons voir facilement que $\mathrm{P}_0^{(i_{k+1})} \in reg(\mathcal{L})$ et $\mathrm{P}_{m_{i_k}}^{(i_k)} \notin \overline{reg(\mathcal{L})}$, et par conséquent $\cup_{i=i_k+1}^{i_{k+1}-1}\mathcal{B}_i \cap \mathcal{L} \neq \emptyset$. Mais $\mathcal{B}$ est une courbe simple, donc la relation ci-dessus implique $\cup_{i=i_k+1}^{i_{k+1}-1}\mathcal{B}_i \cap]\mathrm{P}_0\mathrm{R}_{i_{k+1}}[\neq \emptyset$. Ainsi, il doit exister $i'_0 \in \{i_k+1,\ldots,i_{k+1}-1\}$ tel que $\mathcal{B}_{i'_0} \cap]\mathrm{P}_0\mathrm{R}_{i_{k+1}}[\neq \emptyset$, donc tel que $\mathcal{B}_{i'_0} \cap]\mathrm{R}'_{i_k}\mathrm{R}_{i_{k+1}}[\neq \emptyset$.

Pour conclure la preuve pour le cas (ii), si $i_0 \in \{i_k+1, \ldots, i_{k+1}-1\}$, alors nous obtenons une contradiction, et si $i_0 \in \{i_{k+1}+1, \ldots, i_k-1\}$, alors il existe $i'_0 \in \{i_k+1, \ldots, i_{k+1}-1\}$ ayant les propriétés de i_0 qui nous mènent à une contradiction.

Il est facile à voir que le même raisonnement peut être fait aussi si nous nous situons dans le cas (iv), et ainsi dans ce cas-là également nous obtenons une contradiction.

En conclusion, l'hypothèse

$$\text{il existe } i_0 \in \{1, \ldots, n\} \text{ tel que } \mathcal{P}_{i_0} \subset \mathrm{Int}\Big(\mathcal{E}\big(\bigcup_{i \neq i_0} \mathcal{B}_i\big)\Big) \text{ et } \mathcal{B}_{i_0} \not\subset \mathcal{E}\big(\bigcup_{i \neq i_0} \mathcal{B}_i\big)$$

nous a menés dans tous les cas à une contradiction, donc elle est fausse et ainsi la deuxième propriété de la définition de la pertinence topologique est prouvée et la démonstration de la proposition est achevée. □

Cette proposition nous permet de concevoir un test d'arrêt pour l'algorithme de calcul de la structure topologique de l'enveloppe convexe de la polycourbe, en vérifiant pour chaque B-polygone de courbe de Bézier qui participe à l'enveloppe convexe du polygone de contrôle de la polycourbe si les conditions (III.4.i), (III.4.ii) et (III.4.iii) sont satisfaites, et en subdivisant ce B-polygone si elles ne le sont pas.

III.5.1.2. Pertinence topologique restreinte

Malheureusement, nous ne pouvons pas prouver que l'algorithme utilisant le test d'arrêt ci-dessus converge. Supposons qu'au pas ℓ la courbe $\mathcal{B}_{i_k}$ ne satisfait pas la condition (III.4.i), et supposons aussi que la courbe $\mathcal{B}_{i_{k-1}}$ ne satisfait elle non plus une des trois conditions, et doit être subdivisée. De manière intuitive, il est facile à voir que le diamètre de l'ensemble de points qui "ont le droit" de se trouver dans le demi-plan $\mathrm{DP}(\mathrm{P}_0^{(i_k)}\mathrm{P}_1^{(i_k)}, \mathcal{B}_{i_k})$ decroît au fur à mesure que des subdivisions sont appliquées à la courbe $\mathcal{B}_{i_{k-1}}$, donc il y aura de plus en plus de chances que des sommets autres que ceux de $\mathcal{P}_{i_{k-1}}$ se trouvent dans ce demi-plan.

Comme nous ne sommes pas arrivés à trouver une condition d'arrêt nous permettant de prouver en même temps la pertinence topologique de $\mathcal{E}(\mathcal{P})$ pour $\mathcal{E}(\mathcal{B})$ au moment où cette condition est satisfaite et la convergence de l'algorithme utilisant cette condition, nous nous sommes vus contraints de restreindre la notion de pertinence topologique pour pouvoir donner un algorithme de calcul de la structure topologique de l'enveloppe convexe de la polycourbe.

Définition III.5. – *Soit $\mathcal{B} = \cup_{i=1}^n \mathcal{B}_i$ une polycourbe de Bézier sans tangence intérieure de polygone de contrôle pseudo-simple $\mathcal{P} = \cup_{i=1}^n \mathcal{P}_i$. Pour une courbe de Bézier $\mathcal{B}_i$ de la composition de la polycourbe, soit $\widetilde{\mathcal{B}}_i$ l'union des toutes les courbes $\mathcal{B}_j$ qui proviennent, par subdivision, de la même courbe initiale que $\mathcal{B}_i$: $\widetilde{\mathcal{B}}_i = \bigcup_{\mathcal{B}_j.\mathrm{rang}=\mathcal{B}_i.\mathrm{rang}} \mathcal{B}_j$. De la même manière nous définissons, pour un $i \in \{1, \ldots, n\}$ donné, la ligne polygonale $\widetilde{\mathcal{P}}_i$. Nous dirons que l'enveloppe convexe du polygone de contrôle $\mathcal{P}$ est topologiquement pertinente au sens restreint pour l'enveloppe convexe de la polycourbe si les deux conditions suivantes sont satisfaites :*

(III.5.i) $\mathcal{P}_i \cap \mathcal{Q} \neq \emptyset \Longrightarrow \mathcal{B}_i \cap \mathcal{E} \neq \emptyset$;

(III.5.ii) $\widetilde{\mathcal{P}}_i \subset \mathrm{Int}\left(\mathcal{E}\left(\bigcup_{\widetilde{\mathcal{B}}_j \neq \widetilde{\mathcal{B}}_i} \widetilde{\mathcal{P}}_j\right)\right) \Longrightarrow \widetilde{\mathcal{B}}_i \subset \mathcal{E}\left(\bigcup_{\widetilde{\mathcal{B}}_j \neq \widetilde{\mathcal{B}}_i} \widetilde{\mathcal{B}}_j\right)$

pour tout $i \in \{1, 2, \ldots, n\}$.

Évidemment, la pertinence topologique implique la pertinence topologique au sens restreint et la réciproque n'est pas vraie (figure III.11 : pour la structure courante de la polycourbe, figure (b), la ligne support des polygones $\widetilde{\mathcal{P}}_1 = \mathcal{P}_1$ et $\widetilde{\mathcal{P}}_3 = \mathcal{P}_3 \cup \mathcal{P}_4 \cup \mathcal{P}_5 \cup \mathcal{P}_6$ "touche" $\widetilde{\mathcal{P}}_3$ en un sommet de $\mathcal{P}_4$, tandis que la ligne support des courbes $\widetilde{\mathcal{B}}_1 = \mathcal{B}_1$ et $\widetilde{\mathcal{B}}_3 = \mathcal{B}_3 \cup \mathcal{B}_4 \cup \mathcal{B}_5 \cup \mathcal{B}_6$ "touche" $\widetilde{\mathcal{B}}_3$ en un point de $\mathcal{B}_3$; pour une meilleure visibilité, le polygone $\mathcal{P}_3$ n'a pas été représenté). Nous remarquons aussi que la notion de pertinence topologique au sens restreint n'est pas intrinsèque, car pour une polycourbe donnée il y a plusieurs "structures initiales" possibles. Toutefois, ceci n'est pas gênant pour notre algorithme, qui considère comme structure initiale de la polycourbe celle lue à l'entrée du programme, une polycourbe donnée ayant ainsi une unique structure initiale.

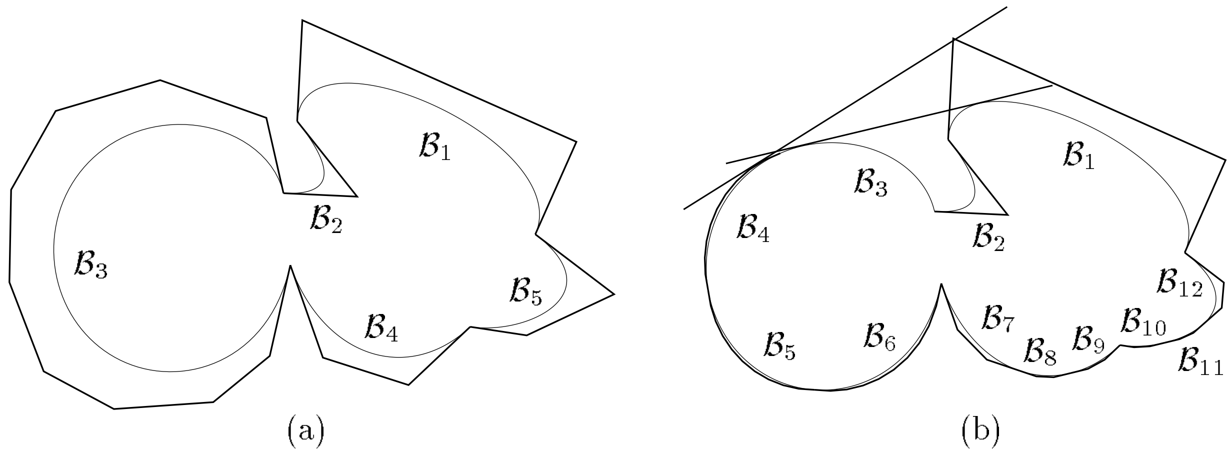

Figure III.11 (a) *Structure initiale de la polycourbe ;* (b) $\mathcal{E}(\mathcal{P})$ *est topologiquement pertinente au sens restreint pour* $\mathcal{E}(\mathcal{B})$ *sans être topologiquement pertinente, car la condition* (III.5.ii) *est satisfaite et* (II.6.ii) *ne l'est pas pour* $i = 3$.

Nous pouvons maintenant énoncer un lemme donnant une condition suffisante pour la pertinence topologique restreinte de l'enveloppe convexe du polygone de contrôle pour celle de la polycourbe :

Lemme III.6. – *En utilisant les mêmes notations, si pour tout* $k \in \mathcal{K}$ *les trois conditions suivantes sont satisfaites, alors l'enveloppe convexe du polygone de contrôle* $\mathcal{E}(\mathcal{P})$ *est topologiquement pertinente au sens restreint pour l'enveloppe convexe de la polycourbe* $\mathcal{E}(\mathcal{B})$ *:*

(III.6.i) *si* P *est un sommet de* $\mathcal{P}$ *tel que* $\mathrm{P} \notin \mathrm{DP}(\mathrm{P}_0^{(i_k)}\mathrm{P}_1^{(i_k)}, \mathcal{B}_{i_k})$*, alors* P *est un sommet de* $\widetilde{\mathcal{P}}_{i_{k-1}}$ *;*

(III.6.ii) *si* P *est un sommet de* $\mathcal{P}$ *tel que* $\mathrm{P} \notin \mathrm{DP}(\mathrm{P}_{m_{i_k}-1}^{(i_k)}\mathrm{P}_{m_{i_k}}^{(i_k)}, \mathcal{B}_{i_k})$*, alors* P *est un sommet de* $\widetilde{\mathcal{P}}_{i_{k+1}}$ *;*

(III.6.iii) *si* $\mathcal{B}_{i_{k-1}}.\mathtt{rang} \neq \mathcal{B}_{i_k}.\mathtt{rang}$ *et* $\mathcal{B}_{i_{k+1}}.\mathtt{rang} \neq \mathcal{B}_{i_k}.\mathtt{rang}$*, alors*

$$\widetilde{\mathcal{B}}_{i_k} \cap \widetilde{\mathcal{P}}_{i_k} \not\subset \mathrm{DP}(\mathrm{PQ}, \mathcal{B}_{i_{k-1}}),$$

où P *est un sommet de* $\mathcal{P}_{i_{k-1}}$ *et* Q *un sommet de* $\mathcal{P}_{i_{k+1}}$ *tels que le segment* [PQ] *soit une arête du polygone frontière de* $\mathcal{E}(\cup_{\alpha\neq k}\mathcal{P}_{i_\alpha})$.

Preuve – La démonstration de ce lemme est en tout point semblable à celle de la proposition III.4, les modifications des conditions(III.4.i), (III.4.ii) et (III.4.iii) correspondant à la différence entre la pertinence topologique et la pertinence topologique restreinte. Pour prouver la propriété (III.5.i) de la définition de la pertinence topologique restreinte (qui est la même que (II.6.i)), nous utilisons la convexité du polygone $\widetilde{\mathcal{P}}_{i_k}$ pour démontrer que $\{\mathrm{P}_0^{(i_k)}, \mathrm{P}_{m_{i_k}}^{(i_k)}\} \not\subset \mathrm{DP}(\mathrm{PQ}, \mathcal{B}_{i_{k-1}})$, avec l'observation que si deux sommets de $\widetilde{\mathcal{P}}_{i_k}$ appartiennent au demi-plan $\mathrm{DP}_{op}(\mathrm{PQ}, \mathcal{B}_{i_{k-1}})$, alors tous les sommets de $\widetilde{\mathcal{P}}_{i_k}$ qui sont "entre" les deux premiers appartiendront eux aussi à ce demi-plan (cette propriété résulte de la convexité du polygone $\widetilde{\mathcal{P}}_{i_k}$, corroborée avec la position que $\widetilde{\mathcal{B}}_{i_k}$ doit avoir par rapport à la droite PQ). La preuve pour la deuxième propriété de la définition de la pertinence topologique restreinte, (III.5.ii), ne présente pas d'aspect nouveau par rapport à celle faite pour la deuxième propriété de la pertinence topologique, (II.6.ii). □

Remarque. Les conditions (III.4.iii) et (III.6.iii) sont en fait équivalentes. Il est évident que (III.4.iii) implique (III.6.iii). Pour l'implication inverse, nous utilisons la remarque, faite dans l'esquisse de preuve ci-dessus, que si deux sommets de $\widetilde{\mathcal{P}}_{i_k}$ appartiennent au demi-plan $\mathrm{DP}_{op}(\mathrm{PQ}, \mathcal{B}_{i_{k-1}})$, alors tous les sommets de $\widetilde{\mathcal{P}}_{i_k}$ qui sont "entre" les deux premiers appartiendront eux aussi à ce demi-plan. Le polygone $\mathcal{P}_{i_k}$ a forcément un sommet dans le demi-plan $\mathrm{DP}_{op}(\mathrm{PQ}, \mathcal{B}_{i_{k-1}})$, soit S_1 ce sommet. De par la condition (III.6.iii), il existe un sommet de $\widetilde{\mathcal{P}}_{i_k}$ qui est en même temps un point de $\widetilde{\mathcal{B}}_{i_k}$ se trouvant dans le demi-plan cité, et soit S_2 ce sommet. Si S_2 est un sommet de $\mathcal{P}_{i_k}$, il ne peut être que $\mathrm{P}_0^{(i_k)}$ ou $\mathrm{P}_{m_{i_k}}^{(i_k)}$ et la preuve est finie. S'il n'est pas un sommet de $\mathcal{P}_{i_k}$, alors tous les sommets de $\widetilde{\mathcal{P}}_{i_k}$ situés entre S_1 et S_2 se trouvent dans le demi-plan $\mathrm{DP}_{op}(\mathrm{PQ}, \mathcal{B}_{i_{k-1}})$, et parmi ces sommets il doit y avoir aussi une des extrémités de $\mathcal{P}_{i_k}$.

Pour pouvoir utiliser les conditions (III.6.i), (III.6.ii) et (III.6.iii) pour la conception d'un test d'arrêt pour notre algorithme, il nous reste à prouver la convergence de l'algorithme qui utilise ce test d'arrêt. Nous devons donc prouver qu'après un nombre fini d'itérations les conditions (III.6.i), (III.6.ii) et (III.6.iii) sont satisfaites par le polygone de contrôle de la polycourbe.

III.5.1.3. Dégénérescence

Malheureusement, l'affirmation ci-dessus n'est toujours pas vraie, à cause de l'existence des cas décrits par la définition suivante :

Définition III.7. – *Soit* $\mathcal{B} = \cup_{i=1}^{n}\mathcal{B}_i$ *une polycourbe de Bézier et soit* $\mathcal{E}$ *la courbe frontière de son enveloppe convexe. Nous dirons que la polycourbe est dégénérée (ou qu'elle présente une dégénéréscence, ou encore que nous avons un cas dégénéré) s'il existe* $i < j < k \in \{1, 2, \ldots, n\}$ *tels que les courbes* $\mathcal{B}_i$, $\mathcal{B}_j$ *et* $\mathcal{B}_k$ *admettent une tangente commune* Δ_{ijk} *et que* $[\mathrm{PQ}] \subset \mathcal{E}$, *où* $\mathrm{P} = \Delta_{ijk} \cap \mathcal{B}_i$ *et* $\mathrm{Q} = \Delta_{ijk} \cap \mathcal{B}_k$.

Nous pouvons voir immédiatement que si une polycourbe est dégénérée, alors aucune des trois conditions ne sera satisfaite, au moins que les paramètres donnant, respectivement, les points de tangence de cette tangente commune sur chacune des courbes de Bézier concernées soient tous de la forme $\alpha/2^\beta$, avec $\alpha \in \{0, 1, \ldots, 2^\beta\}$. Dans ce dernier cas, après $\gamma = \max\{\beta\}$ itérations tous ces points de tangence sont retrouvés sur leurs courbes respectives comme étant l'extrémité d'une (ou deux si $\alpha \notin \{0, 2^\beta\}$) des courbes obtenues par subdivision, et les droites $\mathrm{P}_0^{(i_k)}\mathrm{P}_1^{(i_k)}$ respectives seront confondues, et aussi les droites $\mathrm{P}_{m_{i_k}}^{(i_k)}\mathrm{P}_{m_{i_k}-1}^{(i_k)}$ (figure III.12).

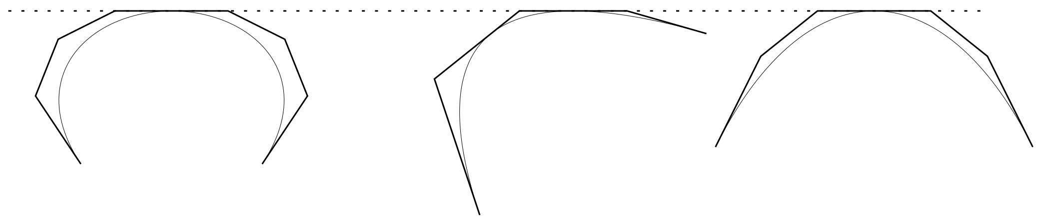

Figure III.12

III.5.1.4. Dégénérescence à ε-près

Si nous avons un cas dégénéré, il est très rare que les points de tangence de la tangente commune aux courbes impliquées soient tous de la forme donnée ci-dessus, et même quand ils le sont, le nombre d'itérations nécessaires pour les retrouver peut être trop élevé. Par conséquent, nous introduisons une nouvelle notion :

Définition III.8. – *Soit $\mathcal{B} = \cup_{i=1}^n \mathcal{B}_i$ une polycourbe de Bézier et ε un réel non-négatif. Nous dirons que $\mathcal{B}$ présente une dégénérescence à ε près s'il existe $i < j < k \in \{1, 2, \ldots, n\}$ tels que $\mathcal{B}_i \cap \mathcal{E} \neq \emptyset$, $\mathcal{B}_k \cap \mathcal{E} \neq \emptyset$ et que l'ensemble $\mathcal{B}_j \cap \mathrm{B}(\mathrm{PQ}, \varepsilon)$ soit un ensemble non vide et connexe, où $\mathrm{P} \in \mathcal{B}_i$ et $\mathrm{Q} \in \mathcal{B}_k$ sont tels que la droite PQ soit la tangente commune des courbes $\mathcal{B}_i$ et $\mathcal{B}_k$ qui réalise l'enveloppe convexe droite de ces deux courbes, et*

$$\mathrm{B}(\mathrm{PQ}, \varepsilon) = \left\{\mathrm{M} \in \mathbb{R}^2 \mid \mathrm{d}(\mathrm{M}, \mathrm{PQ}) \leq \varepsilon\right\}$$

La figure III.13 illustre cette définition.

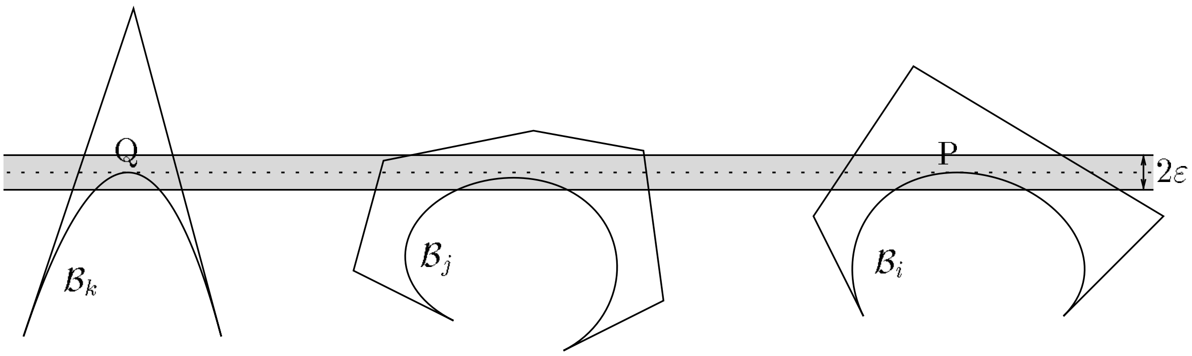

Figure III.13 $\mathcal{B}_j \cap \mathrm{B}(\mathrm{PQ}, \varepsilon)$ *est non-vide et connexe.*

Conformement à la définition, pour ε suffisament grand toute polycourbe sera dégénérée à ε près, mais bien sûr nous allons nous intéresser uniquement aux polycourbes étant dégénérées à ε près pour ε suffisament petit. Dans cette définition, la courbe $\mathcal{B}_j$ peut participer ou pas à la frontière de l'enveloppe convexe de la polycourbe, et il est facile à voir que pour $\varepsilon = 0$ nous retrouvons la définition de la dégénéréscence.

Nous allons étudier le cas où après ℓ itérations une des conditions (III.6.i), (III.6.ii) ou (III.6.iii) n'est pas satisfaite, en trouvant des estimations pour le ε pour lequel la polycourbe est dégénérée à ε près. Un rôle important dans ces estimations est joué par l'angle entre les deux tangentes aux extrémités des courbes des Bézier qui sont concernées par la condition n'étant pas satisfaite, nous devons donc commencer par évaluer la valeur de cet angle après ℓ subdivisions d'une courbe de Bézier.

Soit $\mathcal{B}_0$ une courbe de Bézier de B-polygone $\mathcal{P}_0 = \mathrm{P}_o\mathrm{P}_1\ldots\mathrm{P}_m$. L'angle θ_0 entre les deux tangentes aux extrémités de cette courbe est donné par

$$\theta_0 = angle(\mathrm{P}_1 - \mathrm{P}_0, \mathrm{P}_m - \mathrm{P}_{m-1}) = \left| \int_0^1 \kappa(t)\mathrm{d}t \right| ,$$

où

$$\kappa(t) = \frac{\mathrm{B}'_m(\mathrm{P}_0,\ldots,\mathrm{P}_m;t) \times \mathrm{B}''_m(\mathrm{P}_0,\ldots,\mathrm{P}_m;t)}{\| \mathrm{B}'_m(\mathrm{P}_0,\ldots,\mathrm{P}_m;t) \|^3} \quad \forall t \in [0,1]$$

est la courbure de la courbe de Bézier. Une courbe obtenue à partir de $\mathcal{B}_0$ après ℓ subdivisions sera de la forme $\mathrm{B}_m(\mathrm{P}_0,\ldots,\mathrm{P}_m;[j/2^\ell,(j+1)/2^\ell])$, avec $j \in \{0,1,\ldots,2^\ell-1\}$, et ainsi l'angle entre les tangentes en ses extrémités sera

$$\theta_{\ell j} = \left| \int_{\frac{j}{2^\ell}}^{\frac{j+1}{2^\ell}} \kappa(t)\mathrm{d}t \right| < \left| \int_{\frac{j}{2^\ell}}^{\frac{j+1}{2^\ell}} \kappa_M \mathrm{d}t \right| = \frac{1}{2^\ell} |\kappa_M| ,$$

où $\kappa_M = \max_{t\in[0,1]} \kappa(t)$. Nous pouvons faire cette majoration, car une courbe de Bézier ne pouvant pas être un cercle la courbure ne peut pas être constante. Cette majoration est d'autant plus grossière que la forme du B-polygone $\mathcal{P}_0$ est irrégulière, en comprenant par là que le rapport entre sa plus grande et sa plus petite arête est grand ou qu'il existe un ou plusieurs sommets de ce polygone pour lesquels l'angle intérieur est petit. Ce type d'irrégularité du B-polygone impliquera une forte variation de la courbure pour t allant de 0 à 1, et par conséquent la majoration de la courbure par son maximum faite dans la rélation ci-dessus sera assez bonne pour certaines valeurs de j et très grossière pour d'autres. Nous n'avons pas pu établir une évaluation précise de κ_M en fonction du polygone $\mathcal{P}_0$, ni une meilleure majoration pour l'angle $\theta_{\ell j}$, mais la majoration donnée nous fournit une bonne idée du comportement de cet angle quand ℓ varie (donc quand le nombre de subdivisions varie).

Supposons maintenant qu'après ℓ itérations la condition (III.6.i) n'est pas satisfaite, et soit $k \in \mathcal{K}_0$ et $i \in \mathcal{I} \setminus \mathcal{I}_{k-1}$, $j \in \{0,1,\ldots,m_i\}$ tels que $\mathrm{P}_j^{(i)} \notin \mathrm{DP}(\mathrm{P}_0^{(i_k)}\mathrm{P}_1^{(i_k)},\mathcal{B}_{i_k})$, où $\mathcal{I}_{k-1} = \{h \in \mathcal{I} \mid \widetilde{\mathcal{B}}_h = \widetilde{\mathcal{B}}_{i_{k-1}}\}$. Il est facile à voir que le point $\mathrm{P}_j^{(i)}$ doit appartenir à $\overline{\mathrm{DP}_{op}(\mathrm{P}_0^{(i_k)}\mathrm{P}_1^{(i_k)},\mathcal{B}_{i_k})} \cap \overline{\mathrm{DP}(\mathrm{P}_{m_{i_k}-1}^{(i_k)}\mathrm{P}_{m_{i_k}}^{(i_k)},\mathcal{B}_{i_k})}$, car s'il appartient à la région $\overline{\mathrm{DP}_{op}(\mathrm{P}_0^{(i_k)}\mathrm{P}_1^{(i_k)},\mathcal{B}_{i_k})} \cap \mathrm{DP}_{op}(\mathrm{P}_{m_{i_k}-1}^{(i_k)}\mathrm{P}_{m_{i_k}}^{(i_k)},\mathcal{B}_{i_k})$ nous obtenons une contradiction avec $k \in \mathcal{K}_0$. Le point $\mathrm{P}_{j_{i_{k-1}}+\ell_{i_{k-1}}}^{(i_{k-1})}$, le dernier des sommets de $\mathcal{P}_{i_{k-1}}$ qui est un sommet de $\mathcal{Q}$, doit se trouver lui aussi dans la région $\overline{\mathrm{DP}_{op}(\mathrm{P}_0^{(i_k)}\mathrm{P}_1^{(i_k)},\mathcal{B}_{i_k})} \cap \overline{\mathrm{DP}(\mathrm{P}_{m_{i_k}-1}^{(i_k)}\mathrm{P}_{m_{i_k}}^{(i_k)},\mathcal{B}_{i_k})}$, de par la définition du polygone $\mathcal{Q}$. D'un autre côté, toute droite tangente à la courbe $\mathcal{B}_{i_k}$ aura la pente comprise entre la pente de $P_0^{(i_k)}P_1^{(i_k)}$ et celle de $P_{m_{i_k}-1}^{(i_k)}P_{m_{i_k}}^{(i_k)}$. De ces trois dernières remarques nous pouvons déduire que l'angle formé par la droite $P_{j_{i_k}}P_j^{(i)}$ et $R'_{i_{k-1}}R_{i_k}$, la tangente commune des courbes $\mathcal{B}_{i_{k-1}}$ et $\mathcal{B}_{i_k}$, sera inférieur à θ_{i_k}.

Soit Δ la parallèle à $R'_{i_{k-1}}R_{i_k}$ passant par $P_{j_{i_k}}$. Alors

$$d(P_j^{(i)}, R'_{i_{k-1}}R_{i_k}) \leq d(P_j^{(i)}, \Delta) + d(\Delta, R'_{i_{k-1}}R_{i_k}) \leq d(P_{j_{i_k}}, P_j^{(i)}) \sin\theta_{i_k} + \delta^{H}(\mathcal{P}_{i_k}, [P_0^{(i_k)}P_{m_{i_k}}^{(i_k)}])$$

où $\delta^{H}(\mathcal{P}_{i_k}, [P_0^{(i_k)}P_{m_{i_k}}^{(i_k)}])$ représente évidemment un majorant pour la distance de Hausdorff $\delta^{H}(\mathcal{B}_{i_k}, \mathcal{P}_{i_k})$, qui à son tour majore $d(\Delta, R'_{i_{k-1}}R_{i_k})$. La majoration $d(P_j^{(i)}, \Delta) \leq d(P_{j_{i_k}}, P_j^{(i)}) \sin\theta_{i_k}$ peut être faite grace au fait que l'angle θ_{i_k} est compris entre 0 et $\pi/4$ pour ℓ suffisament grand, intervalle sur lequel la fonction sin est croissante.

Considérons maintenant

$$\varepsilon_M = \inf\{\varepsilon \geq 0 \mid B(R'_{i_{k-1}}R_{i_k}, \varepsilon) \cap \mathcal{B}_i \text{ est non} - \text{vide et connexe}\}$$

Nous avons deux possibilités pour les positions relatives de la courbe $\mathcal{B}_i$ et la droite $R'_{i_{k-1}}R_{i_k}$. Si $\mathcal{B}_i \cap R'_{i_{k-1}}R_{i_k} = \emptyset$, alors $\varepsilon_M = d(\mathcal{B}_i, R'_{i_{k-1}}R_{i_k})$, car suite à la convexité de la courbe $\mathcal{B}_i$ l'ensemble $B(R'_{i_{k-1}}R_{i_k}, \varepsilon) \cap \mathcal{B}_i$ ne peut pas être non-connexe, mais peut être vide. Si $\mathcal{B}_i \cap R'_{i_{k-1}}R_{i_k} \neq \emptyset$, alors $\varepsilon_M = \delta^{H}(\mathcal{B}_i \cap DP_{op}(R'_{i_{k-1}}R_{i_k}, \mathcal{B}_{i_k}), R'_{i_{k-1}}R_{i_k})$, dans ce cas l'ensemble ci-dessus pouvant être non-connexe, mais pas vide. Dans tous les cas, nous avons la majoration

$$\varepsilon_M \leq d(P_j^{(i)}, R'_{i_{k-1}}R_{i_k}) + \delta^{H}(\mathcal{B}_i, \mathcal{P}_i) \leq d(P_j^{(i)}, R'_{i_{k-1}}R_{i_k}) + \delta^{H}(\mathcal{P}_i, [P_0^{(i)}P_{m_i}^{(i)}])$$

Ainsi, pour $\varepsilon > 0$ donné, si

$$d(P_{j_{i_k}}, P_j^{(i)}) \sin\theta_{i_k} + \delta^{H}(\mathcal{P}_{i_k}, [P_0^{(i_k)}P_{m_{i_k}}^{(i_k)}]) + \delta^{H}(\mathcal{P}_i, [P_0^{(i)}P_{m_i}^{(i)}]) \leq \varepsilon$$

et la condition (III.6.i) n'est pas satisfaite, alors nous pouvons conclure que la polycourbe est dégénérée à ε près.

Le raisonnement est tout à fait symétrique si nous supposons qu'après ℓ itérations la condition (III.6.ii) n'est pas satisfaite. Si nous supposons que c'est la condition (III.6.iii) qui n'est pas satisfaite pour $k \in \mathcal{K}_0$, la majoration voulue est encore plus simple :

$$\begin{aligned}\varepsilon_M &\leq \delta^{H}(\mathcal{B}_{i_k}, \mathcal{P}_{i_k}) + \delta^{H}(\mathcal{P}_{i_k}, [P_{j_{i_{k-1}}+\ell_{i_{k-1}}}^{(i_{k-1})} P_{j_{i_{k+1}}}^{(i_{k+1})}]) + \delta^{H}([P_{j_{i_{k-1}}+\ell_{i_{k-1}}}^{(i_{k-1})} P_{j_{i_{k+1}}}^{(i_{k+1})}], [PQ]) \\ &\leq \delta^{H}([P_0^{(i_k)}P_{m_{i_k}}^{(i_k)}], \mathcal{P}_{i_k}) + \delta^{H}(\mathcal{P}_{i_k}, [P_{j_{i_{k-1}}+\ell_{i_{k-1}}}^{(i_{k-1})} P_{j_{i_{k+1}}}^{(i_{k+1})}]) + \delta^{H}([P_{j_{i_{k-1}}+\ell_{i_{k-1}}}^{(i_{k-1})} P_{j_{i_{k+1}}}^{(i_{k+1})}], [PQ])\end{aligned}$$

où les points P et Q sont définis, de la même manière que précédemment, comme étant les points de $\mathcal{B}_{i_{k-1}}$ et respectivement $\mathcal{B}_{i_{k+1}}$ tels que PQ soit la droite d'appui commune de ces deux courbes qui réalise leur enveloppe convexe droite. Pour nous assurer de la dégénérescence à ε près de la polycourbe dans ce cas, il nous suffit donc d'imposer la majoration du membre droit de cette inégalité par ε.

Toutes ces majorations dépendent directement de la structure de la polycourbe après ℓ itérations, et nous ne pouvons pas les utiliser pour déduire combien d'itérations seront nécessaires pour pouvoir conclure, si une des conditions (III.6.i), (III.6.ii) ou (III.6.iii) n'est pas satisfaite, que la polycourbe est dégénérée à ε près, pour un ε donné. Aussi, si la polycourbe est "en position générale", la forme des conditions (III.6.i), (III.6.ii) et

(III.6.iii) ne nous permet pas d'évaluer le nombre d'itérations nécessaires pour que ces trois conditions soient satisfaites. Par conséquent, le coût de notre algorithme dépendra des résultats obtenus et nous pourrons avoir des coûts très différents pour le traitement de deux polycourbes impliquant des données de la même taille, mais ayant des "facteurs de forme" différents.

III.5.1.5. “Localisation” des conditions (III.6.i) et (III.6.ii)

Nous voyons facilement que la condition (III.6.iii) est "locale", elle implique uniquement les polygones $\widetilde{\mathcal{P}}_{i_{k-1}}$, $\widetilde{\mathcal{P}}_{i_k}$ et $\widetilde{\mathcal{P}}_{i_{k+1}}$, tandis que les conditions (III.6.i) et (III.6.ii) sont “globales”, elles impliquent tous les B-polygones des courbes composant la polycourbe. Le résultat suivant nous dit que ces deux conditions peuvent elles aussi être exprimées de manière locale.

Lemme III.9. – *Si les trois conditions suivantes sont satisfaites pout tout $k \in \mathcal{K}$, alors l'enveloppe convexe du polygone de contrôle est topologiquement pertinente au sens restreint pour l'enveloppe convexe de la polycourbe :*

(III.9.i) *si le point* P *est un sommet de* $\cup_{i=i_{k-1}+1}^{i_k-1}\mathcal{P}_i \cup \mathcal{P}_{i_{k-2}}$ *qui se trouve dans le demi-plan* $\mathrm{DP}\big(\mathrm{P}_0^{(i_k)}\mathrm{P}_1^{(i_k)}, \mathcal{B}_{i_k}\big)$*, alors* P *est un sommet de* $\widetilde{\mathcal{P}}_{i_{k-1}}$ *;*

(III.9.ii) *si le point* P *est un sommet de* $\cup_{i=i_k+1}^{i_{k+1}-1}\mathcal{P}_i \cup \mathcal{P}_{i_{k+2}}$ *qui se trouve dans le demi-plan* $\mathrm{DP}\big(\mathrm{P}_{m_{i_k}-1}^{(i_k)}\mathrm{P}_{m_{i_k}}^{(i_k)}, \mathcal{B}_{i_k}\big)$*, alors* P *est un sommet de* $\widetilde{\mathcal{P}}_{i_{k+1}}$ *;*

(III.9.iii) *si* $\mathcal{B}_{i_{k-1}}.\mathtt{rang} \neq \mathcal{B}_{i_k}.\mathtt{rang}$ *et* $\mathcal{B}_{i_{k+1}}.\mathtt{rang} \neq \mathcal{B}_{i_k}.\mathtt{rang}$*, alors*

$$\widetilde{\mathcal{B}}_{i_k} \cap \widetilde{\mathcal{P}}_{i_k} \not\subset \mathrm{DP}\big(\mathrm{PQ}, \mathcal{B}_{i_{k-1}}\big),$$

où P *est un sommet de* $\mathcal{P}_{i_{k-1}}$ *et* Q *un sommet de* $\mathcal{P}_{i_{k+1}}$ *tels que le segment* $[\mathrm{PQ}]$ *soit une arête du polygone frontière de* $\mathcal{E}\big(\cup_{\alpha\neq k}\mathcal{P}_{i_\alpha}\big)$*.*

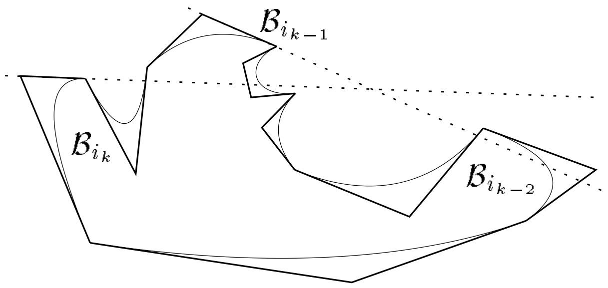

Figure III.14

Il est évident que la condition (III.6.i) implique la condition (III.9.i), et la même rélation existe entre (III.6.ii) et (III.9.ii). La réciproque n'est pas vraie, la figure III.14 présente un cas où la condition (III.9.i) est satisfaite et (III.6.i) pas. La condition (III.9.iii) est identique à (III.6.iii) qui, comme nous avons vu, est équivalente à (III.4.iii).

Preuve du lemme – Nous commencerons cette démonstration par la preuve de la condition (III.5.ii), deuxième de la définition de la pertinence topologique restreinte, preuve qui sera faite par l'absurd. Supposons donc qu'il existe $i \in \{1, 2, \ldots, n\}$ tel que $\mathcal{P}_i \subset \mathrm{Int}\big(\mathcal{E}(\cup_{\alpha\neq i}\mathcal{P}_\alpha)\big)$ et $\mathcal{B}_i \not\subset \big(\mathcal{E}(\cup_{\alpha\neq i}\mathcal{B}_\alpha)\big)$. Comme nous avons déjà vu dans la démonstration de la proposition III.4, et en utilisant les mêmes notations, il existe alors $k \in \mathcal{K}$ et $i_0 \in \{1, 2, \ldots, n\}$ tels que $\mathcal{B}_{i_0} \cap [\mathrm{R}'_{i_k}\mathrm{R}_{i_{k+1}}] \neq \emptyset$. Si $i_0 \in \{i_k + 1, \ldots, i_{k+1} - 1\}$, le raisonnement à faire pour l'obtention d'une contradiction est identique au celui fait dans la preuve de la proposition III.4, nous devons seulement remplacer la région $\mathcal{R}_{i_k}$ par $\mathrm{DP}(\mathrm{P}^{(i_k)}_{m_{i_k}}\mathrm{P}^{(i_k)}_{m_{i_k}-1}, \mathcal{B}_{i_k}) \cap \mathrm{DP}(\mathrm{P}^{(i_k)}_{j_{i_k}+\ell_{i_k}}\mathrm{P}^{(i_{k+1})}_{j_{i_{k+1}}}, \mathcal{B}_{i_k})$ (où les points $\mathrm{P}^{(i_k)}_{j_{i_k}+\ell_{i_k}}$ et $\mathrm{P}^{(i_{k+1})}_{j_{i_{k+1}}}$ sont identifiés par la définition du polygone $\mathcal{Q}$) et la région $\mathcal{R}_{i_{k+1}}$ par $\mathrm{DP}(\mathrm{P}^{(i_{k+1})}_0\mathrm{P}^{(i_{k+1})}_1, \mathcal{B}_{i_{k+1}}) \cap \mathrm{DP}(\mathrm{P}^{(i_k)}_{j_{i_k}+\ell_{i_k}}\mathrm{P}^{(i_{k+1})}_{j_{i_{k+1}}}, \mathcal{B}_{i_k})$. Nous avons aussi vu comment $i_0 \in \{i_{k+1}+1, \ldots, i_k - 1\}$ implique l'existence d'un indice $i'_0 \in \{i_k + 1, \ldots, i_{k+1} - 1\}$ tel que $\mathcal{B}_{i'_0} \cap [\mathrm{R}'_{i_k}\mathrm{R}_{i_{k+1}}] \neq \emptyset$, en menant ainsi à une contradiction, et en conclusion la propriété (III.5.ii) est vérifiée.

Pour la propriété (III.5.i), la démonstration sera aussi faite par l'absurd. En supposant qu'il existe $k \in \mathcal{K}$ tel que $\mathcal{B}_{i_k} \cap \mathcal{E} = \emptyset$ (car l'existence d'un $i \in \{1, 2, \ldots, n\}$ tel que $\mathcal{P}_i \cap \mathcal{Q} \neq \emptyset$ entraîne l'existence d'un $k \in \mathcal{K}$ tel que $i = i_k$), nous montrons qu'il existe $i_0 \in \{1, 2, \ldots, n\}$ tel que $\mathcal{B}_{i_0} \not\subset \mathcal{E}\big(\cup_{k\in\mathcal{K}}\mathcal{B}_{i_k}\big)$ comme nous l'avons fait pour la preuve de la proposition III.4, la seule propriété utilisée étant l'équivalence de (III.9.iii) à (III.4.iii). Nous avons déjà prouvé que ceci impliquera l'existence d'un indice $i \in \{1, 2, \ldots, n\}$ et d'un autre indice $k \in \mathcal{K}$ tels que $\mathcal{B}_i \cap]\mathrm{R}'_{i_k}\mathrm{R}_{i_{k+1}}[\neq \emptyset$, relation qui a mené, dans la preuve ci-dessus pour la deuxième propriété de la définition de la pertinence topologique au sens restreint, à une contradiction. Ainsi, l'hypothèse faite est fausse et la première condition de la définition III.5 est elle aussi vérifiée. □

III.5.2. *Présentation des algorithmes*

Dans cette sous-section, nous décrivons deux algorithmes de calcul de la structure topologique de l'enveloppe convexe d'une polycourbe. Ils représentent deux approches légérement différentes du problème, leurs complexités et précisions respectives offrant le choix en fonction des ressources de l'ordinateur sur lequel une implémentation est envisagée et des résultats attendus.

III.5.2.1. Structures de données spécifiques

Pour exprimer la structure topologique de l'enveloppe convexe que nous aurons calculée, nous avons besoin d'une nouvelle classe, car nous ne nous contenterons pas de fournir l'ensemble $\mathcal{I}$ de la définition II.4, mais essayerons de donner plus de précisions sur la courbe $\mathcal{E}$, frontière de l'enveloppe convexe de la polycourbe. $\mathcal{E}$ est formée par une union d'objets de deux types, d'un côté des parties des courbes de Bézier composant la polycourbe, et de l'autre côté des segments reliant ces parties de courbe. Un tel segment appartient à une droite d'appui commune des courbes dont les parties sont jointes par lui, droite d'appui qui réalise l'enveloppe convexe droite de ces deux courbes.

Nous voulons coder les deux types de composantes de la courbe $\mathcal{E}$ par un seul type de données, les `comp_ec`, et pour ce faire nous avons besoin d'introduire l'`identificateur_point`. Un objet de ce type donne une localisation d'un point d'une des courbes de Bézier composant la polycourbe, par un pointeur à la courbe dans la structure de la polycourbe et un intervalle de paramètres tel que le paramètre correspondant, par la paramétrisation

de Bézier, au point soit compris dans cet intervalle. Il aura donc les champs :
- `courbe` : un pointeur à une courbe de Bézier ;
- `intervalle` : une paire de réels.

Remarque. Nous nous référons ici à la structure **initiale** de la polycourbe, car nous avons vu qu'un polygone de contrôle dont l'enveloppe convexe soit topologiquement pertinente pour celle de la polycourbe ne peut pas être toujours obtenu et nous cherchons donc un polygone de contrôle satisfaisant les conditions de pertinence topologique au sens restreint.

Une `comp_ec` sera ainsi un objet ayant deux champs, `début` et `fin`, chacun étant un `identificateur_point`. Soit $\mathcal{A}$ un `comp_ec` ; si $*(\mathcal{A}.\texttt{début.courbe}) = *(\mathcal{A}.\texttt{fin.courbe})$, alors $\mathcal{A}$ est un morceau d'une des courbes de Bézier de la composition de la polycourbe (celle identifiée par $\mathcal{A}$.`début.courbe`), et si ces deux valeurs sont différentes, alors $\mathcal{A}$ sera un segment de bitangente.

Avec ces définitions, une `env_conv_topologique` est une classe qui hérite la classe méta_polygone<comp_ec>, ayant, en plus des champs `taille` et `éléments`, les trois champs suivants :
- `dégénérescence` : une chaîne de caractères, qui sera "non" si l'algorithme a trouvé un polygone de contrôle de la polycourbe tel que les conditions (III.9.i), (III.9.ii) et (III.9.iii) soient satisfaites, et "oui" dans le cas contraire ;
- `epsilon` : un réel, qui sera -1 si `dégénérescence`="non". Si `dégénérescence`="oui", nous pouvons trouver, comme il a été montré dans le paragraphe III.5.1.3, un ε minimal pour lequel la polycourbe est dégénérée à ε près, et ce champ contiendra la valeur de ε ;
- `localisation` : une liste de listes d'entiers. Si `dégénérescence`="non", cette liste de listes contiendra une seule liste, contenant à son tour un seul entier égal à 0. Si `dégénérescence`="oui", chaque liste d'entiers contiendra les champs `rang` des courbes de Bézier de la composition actuelle de la polycourbe qui sont susceptibles d'avoir (à `epsilon` près) une tangente commune, et la liste de listes contiendra autant de listes d'entiers que l'algorithme a trouvé "d'endroits" sur la polycourbe où une (ou plusieurs) des conditions (III.9.i), (III.9.ii) et (III.9.iii) n'est (ne sont) pas satisfaite(s).

Nous remarquons qu'un `identificateur_point` peut ne pas définir de façon unique un point de la courbe concernée, si `intervalle` n'est pas réduit à un seul point (généralement, il ne le sera pas), mais un `comp_ec` est pourtant défini de façon unique par les objets qui l'encadrent dans la liste `éléments` de l'`env_conv_topologique` à laquelle il appartient. Les intervalles `début.intervalle` et `fin.intervalle` donnent éventuellement des informations supplémentaires, dans le cas le pire tous les deux étant égals à $[0, 1]$. Dans ce cas, tout ce que nous saurons est que pour un `comp_ec` $\mathcal{A}$ avec $*(\mathcal{A}.\texttt{début.courbe}) = *(\mathcal{A}.\texttt{fin.courbe})$ le paramètre correspondant à l'extrémité initiale de $\mathcal{A}$, codée par $\mathcal{A}$.`début`, est inférieur ou égal au paramètre correspondant à l'extrémité finale de $\mathcal{A}$, codée par $\mathcal{A}$.`fin`.

En utilisant ces types de données, il est évident que la liste `ect` sera redondante. En effet, pour tout `i` allant de 0 à `taille`-1, nous avons

$$\texttt{env\_conv\_topologique}[\texttt{i}].\texttt{fin} = \texttt{env\_conv\_topologique}[\texttt{i} + 1].\texttt{début}$$

Nous gardons cette redondance, car il est utile d'avoir la description de chaque élément composant la frontière de l'enveloppe convexe de la polycourbe.

III.5.2.2. Schémas des algorithmes

Tous les types de données nécessaires à nos algorithmes étant définis, nous pouvons maintenant donner leurs structures respectives :

```
Enveloppe_convexe_topologique_1(polycourbe 𝓑, réel ε,
                                    env_conv_topologique 𝒯ℰ)
liste_courbes I
tableau_pointeurs J
entier M
entier i = 0
begin
lecture_données(𝓑)
enveloppe_convexe_polygone_contrôle(𝓑,I)
pertinence_topologique_r(𝓑,I,J)
tant que J≠ ∅ et i ≤ M
     begin
     subdivision(𝓑,J)
     i = i + 1
     enveloppe_convexe_polygone_contrôle(𝓑,I)
     pertinence_topologique_r(𝓑,I,J)
     end
si i > M
     calcul_epsilon(𝓑,J,ε)
sinon
     ε = −1
écriture_resultats(𝓑,I,𝒯ℰ,ε)
end
```

Avec l'algorithme présenté ici, nous fixons le nombre maximal d'itérations, M, et si après M itérations le test d'arrêt (que nous détaillerons) n'est pas satisfait, nous considérons que la polycourbe est dégénérée à ε près, pour une valeur de ε que nous calculons, en utilisant des formules qui seront présentées plus loin. Le nombre d'itérations permis sera choisi en fonction de la machine sur laquelle l'implémentation est faite et des exigences de l'utilisateur concernant la précision.

Nous présentons aussi une variante de cet algorithme où ce n'est pas le nombre maximal d'itérations qui est fixé, mais une valeur maximale pour ε. Cette variante convient quand les ressources de l'ordinateur sur lequel l'implémentation est faite ne sont pas trop limitées, car, comme nous verrons, la complexité de cet algorithme est nettement supérieure à celle de l'algorithme précédent.

```
Enveloppe_convexe_topologique_2(polycourbe 𝓑, réel ε,
                                    env_conv_topologique 𝒯ℰ)
liste_courbes I
tableau_pointeurs J
begin
lecture_données(𝓑,ε)
```

```
    enveloppe_convexe_polygone_contrôle(ℬ,I)
    pert_top_r_eps_près(ℬ,I,ε,J)
    tant que J≠ ∅
        begin
        subdivision(ℬ,J)
        enveloppe_convexe_polygone_contrôle(ℬ,I)
        pert_top_r_eps_près(ℬ,I,ε,J)
        end
    écriture_resultats(ℬ,I,𝒯ℰ,ε)
    end
```

Nous pouvons voir qu'il n'y pratiquement aucune difference entre cet algorithme et celui qui calcule une approximation à ε près, pour un ε donné, de l'enveloppe convexe d'une polycourbe, à part, bien sûr, le test d'arrêt.

Les procédures `enveloppe_convexe_polygone_contrôle` et `subdivision` sont celles décrites dans la première section de ce chapitre. Il nous reste donc à détailler les procédures `pertinence_topologique_r`, `pert_top_r_eps_près` et `calcul_epsilon`.

III.5.2.3. La procédure `pertinence_topologique_r`

Cette procédure est basée sur les conditions (III.9.i), (III.9.ii) et (III.9.iii). Elle a une structure très simple :

```
1.  pertinence_topologique_r(polycourbe ℬ, liste_courbes I,
                              tableau_pointeurs J)
2.  courbe_de_Bézier *P = (premier élément de I).ident_courbe
3.  begin
4.  si non-verifier_en_arrière(ℬ,P, I) ou
       non-verifier_en_avant(ℬ,P, I) ou
       non-verifier_sur_place(ℬ,P, I)
5.       ajouter P à J
6.  si P ne correspond pas au dernier élément de I
7.     faire passer P à l'élément suivant dans l'ensemble I
8.     aller à la ligne 4
9.  end
```

Les procédures `verifier_en_arrière`, `verifier_en_avant` et `verifier_sur_place` sont de type booléen. La premiere retournera "oui" si la condition (III.9.i) est satisfaite par le polygone `*P`, pour le pointeur `P` et l'ensemble `I` donnés, et "non" dans le cas contraire. La deuxième de ces procédures vérifie la condition (III.9.ii), et enfin la troisième vérifie (III.9.iii). Si toutes les trois conditions sont satisfaites, alors l'instruction de la ligne 5 n'est pas exécutée et l'algorithme passe à l'élément suivant de l'ensemble `I`, tant que nous ne sommes pas arrivés au dernier de ses éléments. S'il y a au moins une des trois conditions qui n'est pas satisfaite, alors le pointeur `P` est ajouté au tableau `J` et donc la courbe qu'il designe sera subdivisée à l'itération suivante.

Regardons maintenant quelle est la complexité de cette procédure. Supposons que la polycourbe est composée de n courbes de Bézier ayant le degré maximal d. Le polygone de contrôle de la polycourbe aura donc au plus nd sommets. Le lemme III.9 nous dit que les procédures `vérifier_en_arrière` et `vérifier_en_avant` feront les vérifications de manière "locale". Le coût de la procédure `pertinence_topologique_r` sera alors $\mathcal{O}(nd)$, car pour chaque sommet P du polygone de contrôle de la polycourbe il y a deux tests à faire : si P est un sommet du polygone $\mathcal{P}_i$, avec $i \in \{i_k+1, \ldots, i_{k+1}-1\}$, nous vérifions

$$\mathrm{P} \in \mathrm{DP}(\mathrm{P}^{(i_k)}_{m_{i_k}} \mathrm{P}^{(i_k)}_{m_{i_k}-1}, \mathcal{B}_{i_k}) \quad \text{et} \quad \mathrm{P} \in \mathrm{DP}(\mathrm{P}^{(i_{k+1})}_{0} \mathrm{P}^{(i_{k+1})}_{1}, \mathcal{B}_{i_{k+1}})$$

et si P est un sommet du polygone $\mathcal{P}_{i_k}$, avec $k \in \mathcal{K}$, nous vérifions

$$\mathrm{P} \in \mathrm{DP}(\mathrm{P}^{(i_{k-2})}_{m_{i_{k-2}}} \mathrm{P}^{(i_{k-2})}_{m_{i_{k-2}}-1}, \mathcal{B}_{i_{k-2}}) \quad \text{et} \quad \mathrm{P} \in \mathrm{DP}(\mathrm{P}^{(i_{k+2})}_{0} \mathrm{P}^{(i_{k+2})}_{1}, \mathcal{B}_{i_{k+2}})$$

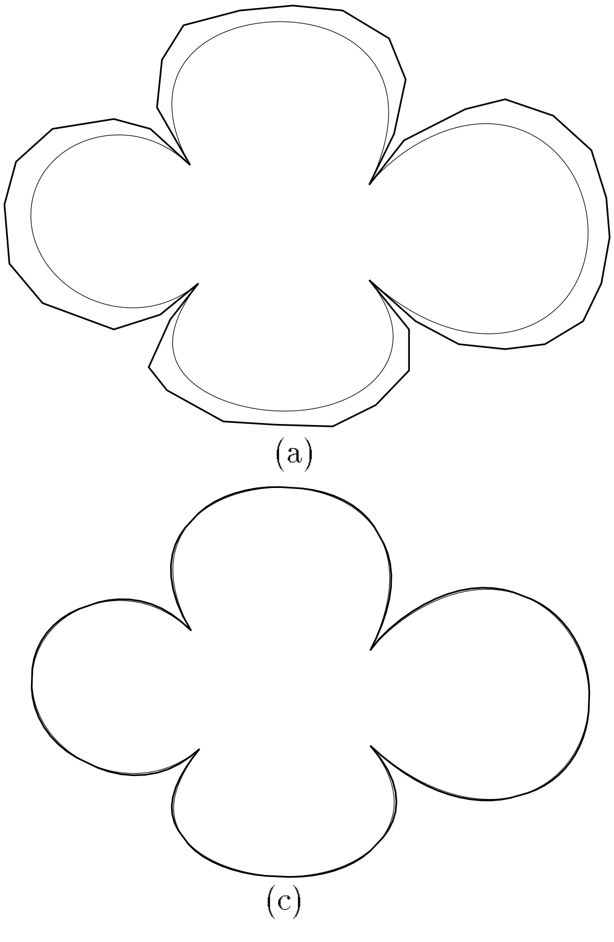

Figure III.15

Les conditions (III.4.i) et (III.4.ii) ne dépendent pas de la structure initiale de la polycourbe, les conditions (III.9.i) et (III.9.ii) si. Pour illustrer la différence entre ces deux paires de conditions, considérons la polycourbe de la figure III.15. Si nous prenons comme structure initiale de cette polycourbe celle de la figure III.15(a), une seule itération suffira pour que les conditions (III.9.i), (III.9.ii) et (III.9.iii) soient satisfaites. Après cette itération, la polycourbe aura la structure donnée figure III.15(b). Si nous considérons cette nouvelle structure comme étant celle initiale de la polycourbe, deux itérations seront nécessaires pour que les conditions (III.9.i), (III.9.ii) et (III.9.iii) soient vérifiées (figure III.15(c) et (d)). En répétant l'opération, onze nouvelles itérations seront cette fois nécessaires pour obtenir les propriétés voulues, et ce procédé ne converge pas. Ceci nous prouve, d'un côté, l'importance de ce que l'algorithme va considérer "la structure initiale" d'une polycourbe donnée, et, d'un autre côté, le fait qu'un algorithme dont le test d'arrêt serait basé sur les conditions (III.4.i), (III.4.ii) et (III.4.iii) ne converge pas (dans ce cas), car la suite d'opérations décrite ci-dessus simule un tel algorithme.

III.5.2.4. La procédure `calcul_epsilon`

Cette procédure donne une valeur ε_0 telle que la polycourbe soit dégénérée à ε_0 près s'il y a au moins une des conditions (III.9.i), (III.9.ii) et (III.9.iii) qui n'est pas satisfaite. ε_0 n'est pas la valeur minimale des ε pour laquelle la polycourbe est dégénérée à ε près, car cette valeur minimale ne peut être calculée uniquement à l'aide du polygone de contrôle.

Supposons que la condition qui n'est pas vérifiée est (III.9.i) et soit $k \in \mathcal{K}$ et $i \in \{1, 2, \dots, n\}$, $j \in \{0, 1, \dots, m_i\}$ tels que

$$\mathrm{P}_j^{(i)} \notin \mathrm{DP}(\mathrm{P}_0^{(i_k)}\mathrm{P}_1^{(i_k)}, \mathcal{B}_{i_k})$$

où i est un indice tel que cette relation contredise la condition (III.9.i). Par conséquent, $\mathcal{B}_i.\mathtt{rang} \neq \mathcal{B}_{i_{k-1}}.\mathtt{rang}$ et nous distinguons deux cas : $i \in \{i_{k-1}+1, \dots, i_k - 1\}$ ou $i = i_{k-2}$.

Si $i \in \{i_{k-1}+1, \dots, i_k-1\}$, nous aurons besoin, pour calculer ε_0, de connaître la distance de Hausdorff entre les courbes $\mathcal{B}_{i_k}$, $\mathcal{B}_{i_{k-1}}$ et $\mathcal{B}_i$ et leurs polygones de contrôle respectifs, distances pour lesquelles nous utiliserons respectivement les majorants $\delta^{\mathrm{H}}(\mathcal{P}_{i_k}, [\mathrm{P}_0^{(i_k)}\mathrm{P}_{m_{i_k}}^{(i_k)}])$, $\delta^{\mathrm{H}}(\mathcal{P}_{i_{k-1}}, [\mathrm{P}_0^{(i_{k-1})}\mathrm{P}_{m_{i_{k-1}}}^{(i_{k-1})}])$ et $\delta^{\mathrm{H}}(\mathcal{P}_i, [\mathrm{P}_0^{(i)}\mathrm{P}_{m_i}^{(i)}])$, comme nous l'avons déjà fait dans la section III.4. L'autre valeur nécessaire au calcul de ε_0 est la distance du point $\mathrm{P}_j^{(i)}$ à la droite $\mathrm{P}_{j_{i_{k-1}}+\ell_{i_{k-1}}}^{(i_{k-1})}\mathrm{P}_{j_{i_k}}^{(i_k)}$. Une fois ces valeurs calculées, nous poserons

$$\varepsilon_0 = \max\{\delta^{\mathrm{H}}(\mathcal{P}_i, [\mathrm{P}_0^{(i)}\mathrm{P}_{m_i}^{(i)}]) + \mathrm{d}(\mathrm{P}_j^{(i)}, \mathrm{P}_{j_{i_{k-1}}+\ell_{i_{k-1}}}^{(i_{k-1})}\mathrm{P}_{j_{i_k}}^{(i_k)}),$$
$$\max\{\delta^{\mathrm{H}}(\mathcal{P}_{i_k}, [\mathrm{P}_0^{(i_k)}\mathrm{P}_{m_{i_k}}^{(i_k)}]), \delta^{\mathrm{H}}(\mathcal{P}_{i_{k-1}}, [\mathrm{P}_0^{(i_{k-1})}\mathrm{P}_{m_{i_{k-1}}}^{(i_{k-1})}])\}\}$$

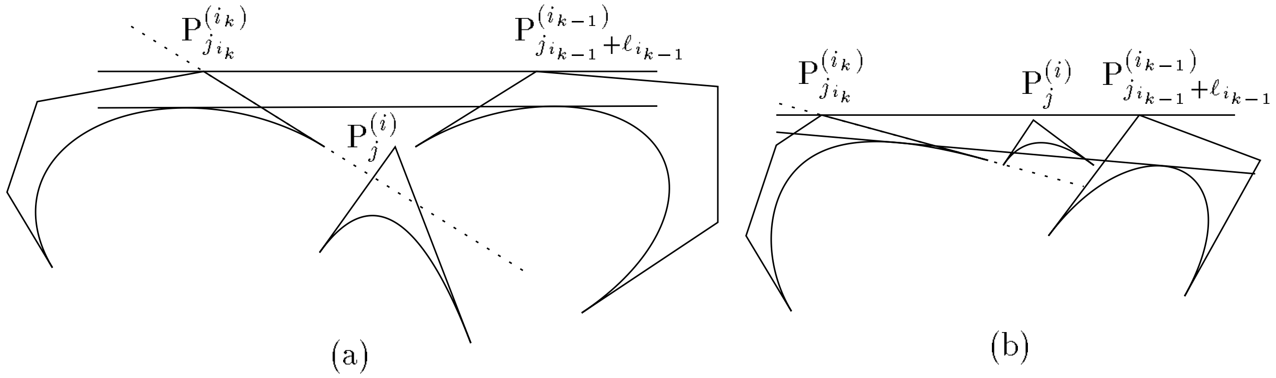

Figure III.16 *Les deux positions possibles de la courbe $\mathcal{B}_i$ par rapport à la droite* $\mathrm{R}'_{i_{k-1}}\mathrm{R}_{i_k}$

Nous pouvons voir sur la figure III.16 que ε_0 est un majorant assez grossier pour le plus petit ε pour lequel la polycourbe est dégénérée à ε près, mais une meilleure évaluation n'a pas pu être obtenue, car nous ne savons pas dans lequel des deux cas de la figure se trouvent les trois courbes.

Si $i = i_{k-2}$, ε_0 dependra de trois valeurs : la distance de Hausdorff entre la courbe $\mathcal{B}_i$ et son polygone de contrôle, que nous majorerons toujours par $\delta^{\mathrm{H}}(\mathcal{P}_i, [\mathrm{P}_0^{(i)}\mathrm{P}_{m_i}^{(i)}])$, la distance du point $\mathrm{P}_j^{(i)}$ à la droite $\mathrm{P}^{(i_{k-1})}_{j_{i_{k-1}}+\ell_{i_{k-1}}} \mathrm{P}^{(i_k)}_{j_{i_k}}$ et la distance de ce point à la droite $\mathrm{P}_0^{(i_k)}\mathrm{P}_1^{(i_k)}$. En connaissant ces valeurs, nous définissons ε_0 par

$$\varepsilon_0 = \max\{\delta^{\mathrm{H}}(\mathcal{P}_i, [\mathrm{P}_0^{(i)}\mathrm{P}_{m_i}^{(i)}]) + \mathrm{d}(\mathrm{P}_j^{(i)}, \mathrm{P}^{(i_{k-1})}_{j_{i_{k-1}}+\ell_{i_{k-1}}} \mathrm{P}^{(i_k)}_{j_{i_k}}),\ \mathrm{d}(\mathrm{P}_j^{(i)}, \mathrm{P}^{(i_{k-1})}_{j_{i_{k-1}}+\ell_{i_{k-1}}} \mathrm{P}^{(i_k)}_{j_{i_k}})\}$$

Dans ce cas aussi, deux positions de la courbe $\mathcal{B}_i$ par rapport à la droite $\mathrm{R}'_{i_{k-1}}\mathrm{R}_{i_k}$ sont possibles, ce qui nous oblige à considérer une majoration grossière de la valeur minimale des ε pour lesquels la polycourbe est dégénérée à ε près.

Les calculs à faire s'il existe $k \in \mathcal{K}$ tel que la condition (III.9.ii) ne soit pas satisfaite par la courbe $\mathcal{B}_{i_k}$ sont symétriques et nous ne les détaillerons plus. Supposons qu'il existe $k \in \mathcal{K}$ tel que $\mathcal{B}_{i_k}$ ne satisfasse pas la condition (III.9.iii). Dans ce cas, nous utilisons

$$\varepsilon_0 = \delta^{\mathrm{H}}(\mathcal{P}_{i_k}, [\mathrm{P}_0^{(i_k)}\mathrm{P}_{m_{i_k}}^{(i_k)}])$$

car si $\mathcal{B}_{i_k} \cap \Delta_{i_{k-1}i_{k+1}} = \emptyset$ (engendrant ainsi une incohérence entre la structure de l'enveloppe convexe de la polycourbe et celle de l'enveloppe convexe du polygone de contrôle), alors cet ε_0 est un majorant pour la distance $\delta^{\mathrm{H}}(\mathcal{B}_{i_k}, \Delta_{i_{k-1}i_{k+1}})$. La droite $\Delta_{i_{k-1}i_{k+1}}$ est la tangente commune des courbes $\mathcal{B}_{i_{k-1}}$ et $\mathcal{B}_{i_{k-1}}$ qui réalise l'enveloppe droite de ces deux courbes.

Nous ne donnons pas le schéma de la procédure `calcul_epsilon`, car elle ne contient pas autre chose que les calculs décrits ci-dessus. Il est facile à voir que la complexité d'un tel calcul est $\mathcal{O}(d)$, d étant le degré maximal des courbes de Bézier composant la polycourbe.

III.5.2.5. La procédure `pert_top_r_eps_près`

La structure de cette procédure est la suivante :

```
1.    pert_top_r_eps_près(polycourbe ℬ, réel ε, liste_courbes I,
                          tableau_pointers J)
2.    courbe_de_Bézier *P = (premier élément de I).ident_courbe
3.    begin
4.    si non-verifier_en_arrière(ℬ,P,I)
5.        si maj_epsilon_1(P,I, ℬ)≥ ε
6.            begin
7.            ajouter P à J
8.            aller à la ligne 22
9.            end
10.   si non-verifier_en_avant(ℬ,P,I)
11.       si maj_epsilon_2(P, I,ℬ)≥ ε
12.           begin
13.           ajouter P à J
14.           aller à la ligne 22
15.           end
16.   si non-verifier_sur_place(ℬ,P,I)
17.       si maj_epsilon_3(P, I,ℬ)≥ ε
18.           begin
19.           ajouter P à J
```

```
20.             aller à la ligne 22
21.             end
22.     si P ne correspond pas au dernier élément de I
23.         begin
24.         faire passer P à l'élément suivant dans l'ensemble I
25.         aller à la ligne 4
26.         end
27.     end
```

Nous remarquons que cette procédure est, structurellement, assez différente de `pertinence_topologique_r`, mais le principe est le même. Les procédures `verifier_en_arrière`, `verifier_en_avant` et `verifier_sur_place` sont celles appelées aussi par `pertinence_topologique_r`. La procédure `maj_epsilon_1` calcule, en utilisant la formule donnée dans le paragraphe précédent, la valeur de ε_0, en sachant que la courbe $\mathcal{B}_{\mathrm{i}}$ ne satisfait pas la condition (III.9.i), car `verifier_en_arrière` a retourné, pour cet indice, "non". De la même manière, `maj_epsilon_2` calcule la valeur de ε_0 quand la courbe $\mathcal{B}_i$ ne satisfait pas la condition (III.9.ii) et `maj_epsilon_3` donne la valeur de ε_0 quand $\mathcal{B}_i$ ne vérifie pas la condition (III.9.iii). Si une des conditions (III.9.i), (III.9.ii) et (III.9.iii) n'est pas satisfaite et en plus le ε_0 correspondant à cette condition est supérieur ou égal à ε, la courbe $\mathcal{B}_i$ doit être subdivisée, car l'enveloppe convexe du polygone de contrôle peut ne pas être topologiquement pertinente pour celle de la polycourbe, à cause de la condition non vérifiée, et la polycourbe peut ne pas être dégénérée à ε près, car ε_0 calculé n'est pas inférieur à ε.

Les procédures `maj_epsilon_i`, pour `i` $\in \{1,2,3\}$, ont chacune la complexité $\mathcal{O}(d)$. Nous pouvons remarquer que si, pour $k \in \mathcal{K}$ donné, il existe plusieurs sommets $\mathrm{P}_j^{(i)}$ tels que la condition (III.9.i) soit contredite, nous obtenons une valeur de ε_0 pour chacun de ces sommets. Ces valeurs ne sont généralement pas égales, mais elles sont toutes des majorations pour le minimum de l'ensemble des ε pour lesquels la polycourbe est dégénérée à ε près. Notre procédure devrait chercher la plus petite valeur de ε_0, pour réduire le nombre d'itérations à faire dans `Enveloppe_convexe_topologique_1` et `Enveloppe_convexe_topologique_2`, mais à cause du coût que cette opération impliquerait pour les procédures de calcul de ε_0 il est plus pratique de prendre la valeur donnée par un tel sommet $\mathrm{P}_j^{(i)}$ quelconque. Nous voyons tout de suite qu'en utilisant cette méthode de calcul, la procédure `pert_top_r_eps_près` a la même complexité que la procédure `pertinence_topologique_r`, donc $\mathcal{O}(nd)$.

III.5.2.6. Le coût des procédures `Enveloppe_convexe_topologique_i, i=1,2`

Pour calculer la complexité de la procédure `Enveloppe_convexe_topologique_1`, il faut sommer les coûts des procédures `enveloppe_convexe_polygone_contrôle` (ligne 7), `pertinence_topologique` (ligne 8), `calcul_epsilon` (ligne 17) et du cycle des lignes 9-15. Mais nous remarquons que cette procédure a exactement la même structure que `Enveloppe_convexe_approchée`, à part les lignes 16-19, et qu'en plus les procédures appelées sont elles aussi les mêmes. La seule exception est constituée par la procédure dist_inf_eps appelée dans Enveloppe_convexe_approchée, qui est ici remplacée par pertinence_topologique_r. Mais comme ces deux procédures ont la même complexité, cette 'substitution' n'affecte pas le coût total du programme. Le coût qui nous intéresse sera, en utilisant les notation du sous-chapitre précédent, $\mathcal{O}\big(n_0 d^2 M(M+1)\big)$, auquel il faut ajouter le coût de calcul_epsilon. Nous avons vu que le coût du calcul d'un ε_0 est $\mathcal{O}(d)$ et, pour des raisons déjà mentionnées, nous aurons au plus $6n_0$ valeurs de ε_0 à calculer (au plus 3 valeurs, correspondant chacune à la non-vérification d'une des conditions du lemme III.9, pour chaque courbe de Bézier dont l'indice se trouve dans l'ensemble $\mathcal{I}_0$, dont le cardinal est au plus $2n_0$) et il faudra après calculer le maximum de toutes ces valeurs, et par conséquent la procédure calcul_epsilon coûtera, dans le cas le pire, $\mathcal{O}(n_0 d)$. Ainsi, la complexité totale de Enveloppe_convexe_topologique_1 est $\mathcal{O}\big(n_0 d^2 M(M+1)\big)$.

Le même raisonement peut se faire pour le calcul de la complexité de Enveloppe_convexe_topologique_2, qui sera exactement la même que celle de Enveloppe_convexe_approchée, car pert_top_r_eps_près à le même coût que dist_inf_eps. Par contre, dans ce cas aucune évaluation du nombre d'itérations nécessaire pour que la procédure pert_top_r_eps_près construise un ensemble J vide n'est possible, et par conséquence le coût sera $\mathcal{O}\big(n_0 d^2 k_T(k_T+1)\big)$, où k_T est le nombre total (inconnu) d'itérations.

III.5.2.7. L'algorithme implémenté

Nous avons présenté deux schémas d'algorithme de calcul de la structure topologique de l'enveloppe convexe d'une polycourbe de Bézier sans tangence intérieure. Le premier de ces deux algorithmes, Enveloppe_convexe_topologique_1, a l'avantage d'un coût contrôlable par l'utilisateur et l'inconvénient de retourner un résultat qui peut être assez éloigné du résultat désiré. Le deuxième algorithme, Enveloppe_convexe_topologique_2, a les caractéristiques opposées, le résultat peut être contrôlé par l'utilisateur, mais le coût échappe à tout contrôle.

Généralement, la qualité du résultat obtenu est la plus importante, la complexité de l'algorithme utilisé passant en deuxième. Par conséquent, c'est le deuxième des algorithme proposés qui sera privilégié pour une implémentation. En regardant l'étude de coût faite le paragraphe précédent, nous voyons que pour un nombre égal d'itérations les algorithmes Enveloppe_convexe_topologique_1 et Enveloppe_convexe_topologique_2 ont la même complexité. Mais nous pouvons facilement remarquer qu'il y a plus d'opérations à faire pour une itération de Enveloppe_convexe_topologique_2 que pour l'itération correspondante de Enveloppe_convexe_topologique_1, les opérations supplémentaires étant nécessaires au calcul de ε_0. Dans certains cas, il peut y avoir même deux fois plus d'opérations à faire pour Enveloppe_convexe_topologique_2 que pour Enveloppe_convexe_topologique_1.

Pour "alléger" le coût du programme que nous proposons, nous allons implémenter un algorithme combinant les deux décrits précédement. Il est clair que le calcul des ε_0 n'est pas important pendant les premières itérations, car les valeurs des ε_0 dans les phases du début du déroulement de l'algorithme sont, généralement, relativement grandes. Nous pouvons donc exécuter un certain nombre d'itérations sans nous poser le problème de la dégénerescence à ε près. Ainsi, les quinze premières lignes de `Enveloppe_convexe_topologique_1` se recopient identiquement dans le schéma de l'algorithme que nous décrivons ici, et que nous appellerons `Enveloppe_convexe_topologique`. Si après M itérations (nous utiliserons $M = 15$) l'enveloppe convexe du polygone de contrôle n'est toujours pas topologiquement pertinente pour celle de la polycourbe, nous avons des bonnes raisons de supposer que la polycourbe est dégénérée à ε près pour une valeur de ε petite, et nous allons donner le ε maximal admissible. La procédure finit donc avec les lignes 7-14 de `Enveloppe_convexe_topologique_2`. La remarque à faire est que les deux procédures construisant l'ensemble `J` qui est comparé à l'ensemble vide pour le test d'arrêt, `pertinence_topologique_r` et `pert_top_r_eps_près`, sont appelées par cet algorithme.

III.6. L'enveloppe convexe "mixte"

Il est parfois utile de trouver pour une polycourbe donnée l'enveloppe convexe approchée et en même temps la description de l'enveloppe convexe topologique. Ces deux enveloppes convexes étant, comme décrit dans les deux sections précédentes, des objets de natures différentes, il n'est pas possible de fournir un objet qui ait à la fois leurs propriétés respectives. Nous pouvons, par contre, calculer un polygone de contrôle de la polycourbe dont l'enveloppe convexe soit topologiquement pertinente au sens restreint pour celle de la polycourbe, tout en n'étant pas séparée de celle-là par une distance de Hausdorff supérieure à un ε donné.

Soit $\mathcal{B}$ une polycourbe de Bézier et ε un réel positif. En utilisant les notations des chapitres précédents, pour que le polygone de contrôle courant de $\mathcal{B}$ puisse donner en même temps une approximation de $\mathcal{E}(\mathcal{B})$ avec une erreur inférieure à ε et la structure topologique de $\mathcal{E}(\mathcal{B})$ les tableaux `J'` et `J"`, donnés respectivement par les procédures `dist_inf_eps` et `pertinence_topologique_r`, doivent être tous les deux vides. Le schéma d'un algorithme calculant le polygone de contrôle voulu pourrait alors être le suivant :

```
Enveloppe_convexe_mixte(polycourbe ℬ, réel ε, polygone_contrôle 𝒫)
liste_courbes I
tableaux_pointeurs J', J"
begin
lecture_données(ℬ,ε)
enveloppe_convexe_polygone_contrôle(ℬ,I)
dist_inf_eps(ℬ,I,ε,J')
pertinence_topologique_r(ℬ,I,J")
tant que J'∪J"≠ ∅
     begin
     subdivision(ℬ,J'∪J")
     enveloppe_convexe_polygone_contrôle(ℬ,I)
     dist_inf_eps(ℬ,I,ε,J')
     pertinence_topologique_r(ℬ,I,J")
     end
```

```
16.    écriture_resultats(ℬ,I,𝒫)
17.    end
```

À chaque itération, chacun des B-polygones $\mathcal{P}_i$ identifiés par le tableau `I`, donc qui participe à la frontière de l'enveloppe convexe du polygone de contrôle de la polycourbe, subit les deux types de test, ceux nécessaires pour assurer une approximation avec une erreur inférieure à la valeur de ε donnée et qui demandent le calcul de $\delta^{\mathrm{H}}(\mathcal{P}_i, [\mathrm{P}_0^{(i)}\mathrm{P}_{m_i}^{(i)}])$, et ceux donnés par les conditions (III.9.i), (III.9.ii) et (III.9.iii). Cet algorithme minimise le nombre d'itérations à faire, mais pas le nombre d'opérations.

En effet, dans la plupart des cas le tableau `J'` contiendra, au moins pendant les premières itérations, des pointeurs à toutes les courbes de Bézier composant la polycourbe identifiées par le tableau `I` et ayant le caractère associé "y" (voir le paragraphe III.3.1), qui sont toutes les courbes de Bézier données par l'ensemble $\widetilde{\mathcal{I}}_0$ de la formule (III.1). Les exceptions à cette règle peuvent être constituées par les polycourbes pour lesquelles il existe une grande disproportion (des ordres de grandeur) entre

$$\max_{i\in\widetilde{\mathcal{I}}_0}\Big(\max_{j\in\{1,\ldots,m_i\}} \mathrm{d}(\mathrm{P}_{j-1}^{(i)}, \mathrm{P}_j^{(i)})\Big) \qquad \text{et} \qquad \min_{i\in\widetilde{\mathcal{I}}_0}\Big(\max_{j\in\{1,\ldots,m_i\}} \mathrm{d}(\mathrm{P}_{j-1}^{(i)}, \mathrm{P}_j^{(i)})\Big)$$

Tant que le tableau `J'` designe les mêmes courbes de Bézier que l'ensemble $\widetilde{\mathcal{I}}_0$, nous aurons évidemment `J"`$\subset$`J'`, car nous ne subdivisons jamais une courbe dont l'indice ne se trouve pas dans $\widetilde{\mathcal{I}}_0$. La conséquence de cette inclusion est qu'à chaque itération l'algorithme exécute $3\,\mathrm{card}(\widetilde{\mathcal{I}}_0)$ tests inutiles, correspondant à la procédure `pertinence_topologique_r`.

Pour ces raisons, l'algorithme qui a été implémenté correspond au schéma suivant :

```
1.    Enveloppe_convexe_mixte(polycourbe ℬ, réel ε, polygone_contrôle 𝒫)
2.    begin
3.    enveloppe_convexe_approchée(ℬ, ε)
4.    enveloppe_convexe_topologique(ℬ)
5.    polygone_de_contrôle(ℬ, 𝒫)
6.    end
```

Les procédures des lignes 3 et 4 sont les deux algorithmes amplement décrits dans les deux chapitres précédents. La procédure de la ligne 5 construit le polygone de contrôle d'une polycourbe donnée, elle est brièvement présentée dans le paragraphe III.3.2.2. Ce nouvel algorithme traite aussi les cas dégénérés, car l'algorithme de la procédure `enveloppe_convexe_topologique` le fait. Remarquons que les itérations effectuées par `enveloppe_convexe_approchée` sont prises en compte par le compteur d'itérations du premier cycle de l'algorithme `enveloppe_convexe_topologique` (qui est le cycle des lignes 10-16 de `enveloppe_convexe_topologique_1`).

Si nous supposons que l'algorithme `enveloppe_convexe_topologique` a décelé une possibilité de dégénérescence pour la polycourbe $\mathcal{B}$, il est très facile à voir que le ε_{min} minimal pour lequel nous avons une dégénérescence à ε-près de $\mathcal{B}$ est inférieur à la valeur que nous avons donnée pour le calcul de l'enveloppe convexe approchée à ε-près. Cette observation est intéressante, car les valeurs ε_0 calculées par les procédures `vérifier_en_arrière`, `vérifier_en_avant` et `vérifier_sur_place` sont, comme nous l'avons remarqué, des majorations très grossières de ε_{min}.

III.7. Résultats numériques

Les algorithmes décrits dans les sections précédentes ont été implémentés en C++, en utilisant, pour l'interface graphique, la bibliothèque de fonctions LEDA. Dans cette section nous présentons les résultats obtenus avec ces programmes. Avant de commencer, précisons que l'unité de mesure (utilisée pour l'évaluation de toutes les distances) est choisie telle que 1 corresponde à 1 mm sur un écran de 17 pouces en résolution 1280×1020.

Les programmes ont été appliquées à un échantillon de 85 polycourbes et nous avons calculé des moyennes pour diverses valeurs qui nous ont semblé rélévantes pour le fonctionnement et les complexités des algorithmes. Le tableau suivant présente les moyennes pour :

- n, le nombre de courbes de Bézier composant la polycourbe ;
- m, le nombre de sommets du polygone de contrôle ;
- κ_T, le nombre d'itérations nécessaires pour que les tests d'arrêt respectifs soient satisfaits ;
- n_f/n_0, le taux de croissance (en courbes de Bézier) de la polycourbe, c'est-à-dire le rapport entre la taille de la polycourbe initiale et celle de la polycourbe satisfaisant le test d'arrêt des algorithmes respectifs ;
- m_f/m_0, le taux de croissance (en sommets) du polygone de contrôle.

	Polycourbe initiale	Env. conv. topologique	Env. conv. approchée		
			$\varepsilon = 10^{-1}$	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-3}$
n	$12,63$	$37,74$	$44,29$	$57,55$	$70,59$
m	$42,56$	$148,87$	$177,235$	$233,67$	$288,35$
κ_T	—	$6,11$	$4,7$	$6,33$	$7,88$
n_f/n_0	—	$2,99$	$3,51$	$4,656$	$5,59$
m_f/m_0	—	$3,62$	$5,11$	$6,7$	$8,25$

La première remarque que nous pouvons faire est que l'algorithme de calcul de l'enveloppe convexe approchée génère une croissance nettement plus importante de la polycourbe et du polygone de contrôle que l'algorithme calculant la structure topologique de cette enveloppe convexe, même quand le nombre d'itérations nécessaires pour l'enveloppe convexe topologique est largement plus grand que le nombre d'itérations appliquées pour l'enveloppe convexe approchée, comme c'est le cas quand $\varepsilon = 10^{-1}$. Ceci s'explique par le fait que les subdivisions se font, dans le cas de l'enveloppe convexe topologique, de manière bien plus sélective que dans le cas de l'enveloppe convexe approchée, où toutes les courbes designées par l'ensemble $\widetilde{\mathcal{I}}_0$ sont subdivisées.

Une autre observation est que n_f/n_0, qui dans l'étude de coût faite dans les paragraphes précédents avait été évalué à $2\kappa_T$, est légérement supérieur à $(2/3)\kappa_T$ pour l'enveloppe convexe approchée et inférieur à $\kappa_T/2$ pour l'enveloppe convexe topologique. Ainsi, n_f est toujours $\mathcal{O}(n_0)$, mais le facteur de proportionnalité est bien plus petit. Une autre remarque interessante est que le taux de croissance du polygone de contrôle est supérieure à celle de la polycourbe. Si toutes les courbes de Bézier composant une polycourbe avaient le même degré, et ceci pour chacune des polycourbes de notre échantillon, les deux taux de croissance auraient été égaux. L'inégalité existant entre eux signifie que les courbes qui ont été subdivisées sont, généralement, de degré supérieur aux celles qui ne l'ont pas été. Même ainsi, le taux de croissance du polygone de contrôle est plutôt de l'ordre de κ_T que de l'ordre de $2\kappa_T$, pour l'enveloppe convexe approchée, et de l'ordre de $\kappa_T/2$ pour l'enveloppe convexe topologique. Ceci se répercutera sur la complexité des algorithmes, car elle dépend directement de la taille du polygone de contrôle.

Il faut aussi mentionner que le coût, aussi bien théorique que pratique, de l'algorithme de calcul de l'enveloppe convexe approchée est considerablement réduit par l'utilisation du lemme III.3. En effet, si nous cherchions une approximation polygonale de l'enveloppe convexe, sans remplacer tout B-polygone dont tous les sommets sont des sommets de $\mathcal{Q}$ par la courbe de Bézier correspondante, mais en subdivisant cette courbe jusqu'à ce que la distance de Hausdorff entre elle et son B-polygone soit inférieure à la valeur ε donnée, alors la complexité théorique de l'algorithme serait $\mathcal{O}(n_0 d^2 2^{\kappa_T})$. Théoriquement, le taux de croissance de la polycourbe serait 2^{κ_T} et nous n'avons pas implémenté cette variante de l'agorithme pour pouvoir donner les valeurs obtenues pratiquement, mais nous estimons qu'elle serait à peu près de l'ordre de $2^{\kappa_T/2}$.

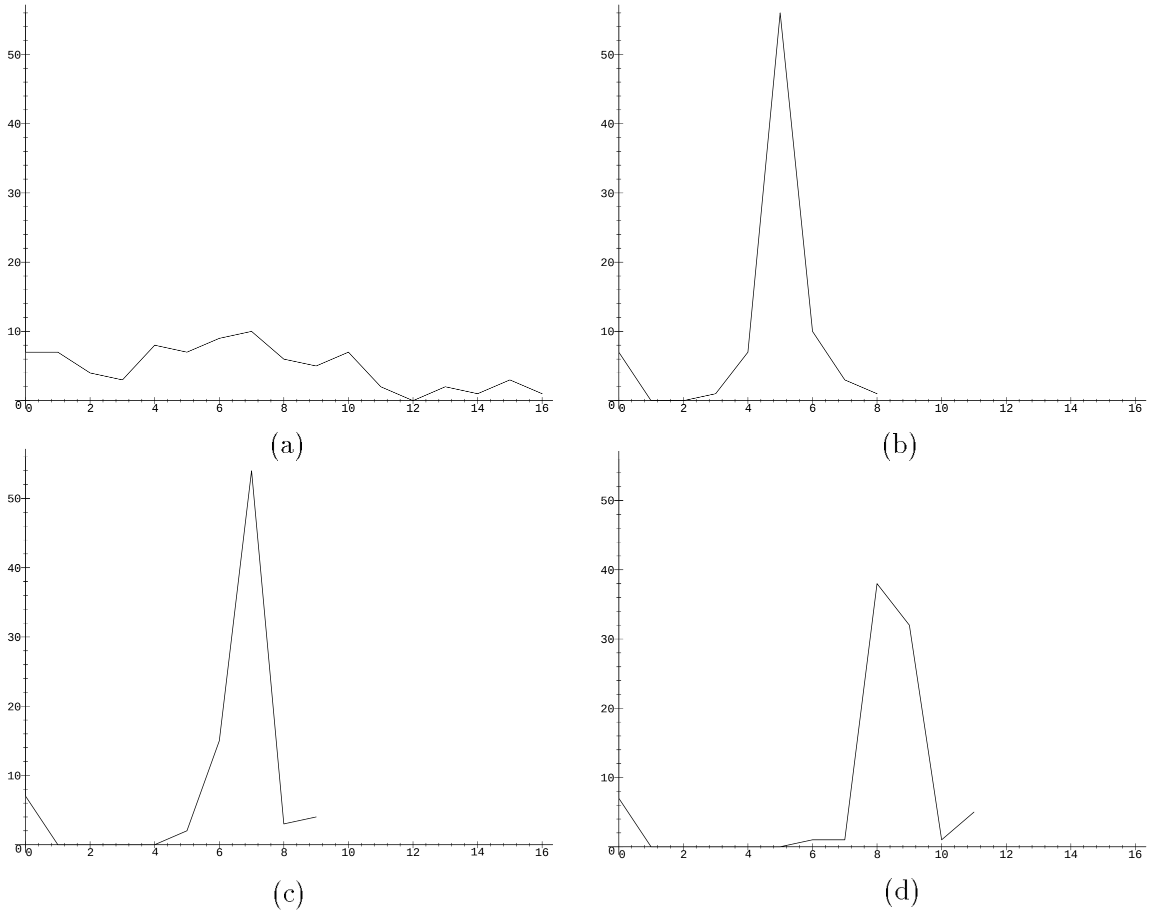

Figure III.17 *Répartition des polycourbes de l'échantillon par nombre d'itérations effectuées :* (a) *enveloppe convexe topologique ;* (b), (c), (d) *enveloppe convexe approchée respectivement pour* $\varepsilon = 10^{-1}$, $\varepsilon = 10^{-2}$, $\varepsilon = 10^{-3}$

Le nombre maximal d'itérations appliquées par le programme calculant l'enveloppe convexe topologique a été, sur notre échantillon, 16, car il existe une polycourbe pour laquelle une possibilité de dégénérescence a été détectée (remarquons qu'après 16 itérations, la valeur de dégénérescence ε_0 calculée pour cette polycourbe était inférieure à 10^{-4}). Les nombres maximaux d'itérations appliquées par le programme de calcul de l'enveloppe convexe approchée ont été, respectivement pour $\varepsilon = 10^{-1}$, $\varepsilon = 10^{-2}$ et $\varepsilon = 10^{-3}$, 8, 9 et 11. La figure III.17 présente les graphes de répartition des polycourbes par nombre d'itérations effectuées.

Si pour l'algorithme de calcul de l'enveloppe convexe approchée les graphes présentent une concentration très marquée à la proximité de la valeur moyenne du nombre d'itérations appliquées, ceci n'est pas le cas pour l'algorithme calculant l'enveloppe convexe topologique. Ce phénomène s'explique par la nature du résultat cherché. Les polycourbes de l'échantillon ont des dimensions des arêtes comparables, ce qui fait qu'elle doivent toutes subir plus ou moins le même nombre d'itérations pour que l'expression majorant la distance de Hausdorff entre une courbe de Bézier et son B-polygone que nous avons utilisée dans nos tests d'arrêt devienne inférieure à la valeur ε donnée, ce qui explique les pics des figures III.17(b), (c) et (d). Pour la relative uniformité de la répartition dans le cas de l'enveloppe convexe topologique nous ne sommes pas arrivés à trouver une explication dans les tests d'arrêt utilisés, sinon qu'ils n'influencent pas cette répartition, qui sera, par conséquent, plus ou moins aléatoire (l'échantillon testé étant alors trop petit pour que nous ayons une image complète du phénomène).

Nous remarquons que la répartition des polycourbes par nombre d'itérations effectuées nous donne très peu d'indications sur le nombre de subdivisions que les courbes de Bézier composant les polycourbes de l'échantillon subissent. Ceci est dû au fait que les polycourbes n'ont pas toutes la même taille, mais surtout au fait que pour une polycourbe donnée nous ne devons généralement pas subdiviser à chaque itération toutes les courbes de Bézier la constituant. La figure III.18 présente les graphes de repartition des courbes de Bézier par nombre de subdivisions appliquées pour les deux algorithmes (cette fois, le nombre de courbes est exprimé en pourcents du nombre total de courbes, qui avoisine 900).

Une précision doit être faite sur la notion de "courbe de Bézier subdivisée k fois".

Définition III.10. – *Soit $\mathcal{B}$ une polycourbe et soit $\mathcal{B}_i$ une des courbes de Bézier de sa composition initiale. Nous dirons que $\mathcal{B}_i$ a été subdivisée k fois si parmi les courbes de Bézier de la composition finale de la polycourbe il exite une, appelons-la $\mathcal{B}_{i_f}$, telle que $\mathcal{B}_{i_f}.\mathtt{rang} = i$ et $\mathcal{B}_{i_f}.\mathtt{profondeur} = k$.*

Sur ces graphes, nous pouvons remarquer que près de la moitié des courbes de Bézier de l'échantillon ne sont jamais subdivisées, soit parce qu'elles n'ont aucune importance pour l'enveloppe convexe, soit parce que nous pouvons déterminer de manière exacte quelles parties d'elles se trouvent sur la frontière de l'enveloppe convexe (cas des courbes identifiées par les ensembles $\mathcal{I}_1$ et $\mathcal{I}_2$ du lemme III.2 et de celles dont tous les points de contrôle sont des sommets de $\mathcal{Q}$). Ces courbes-là mises à part, nous voyons que les graphes correspondant à l'algorithme de calcul de l'enveloppe convexe approchée ont la même alure que celles de la figure III.16 respectivement pour les mêmes valeurs de ε, et les pics de III.17(b), (c) et

(d) s'expliquent de la même manière que ceux de III.16(b), (c) et (d).

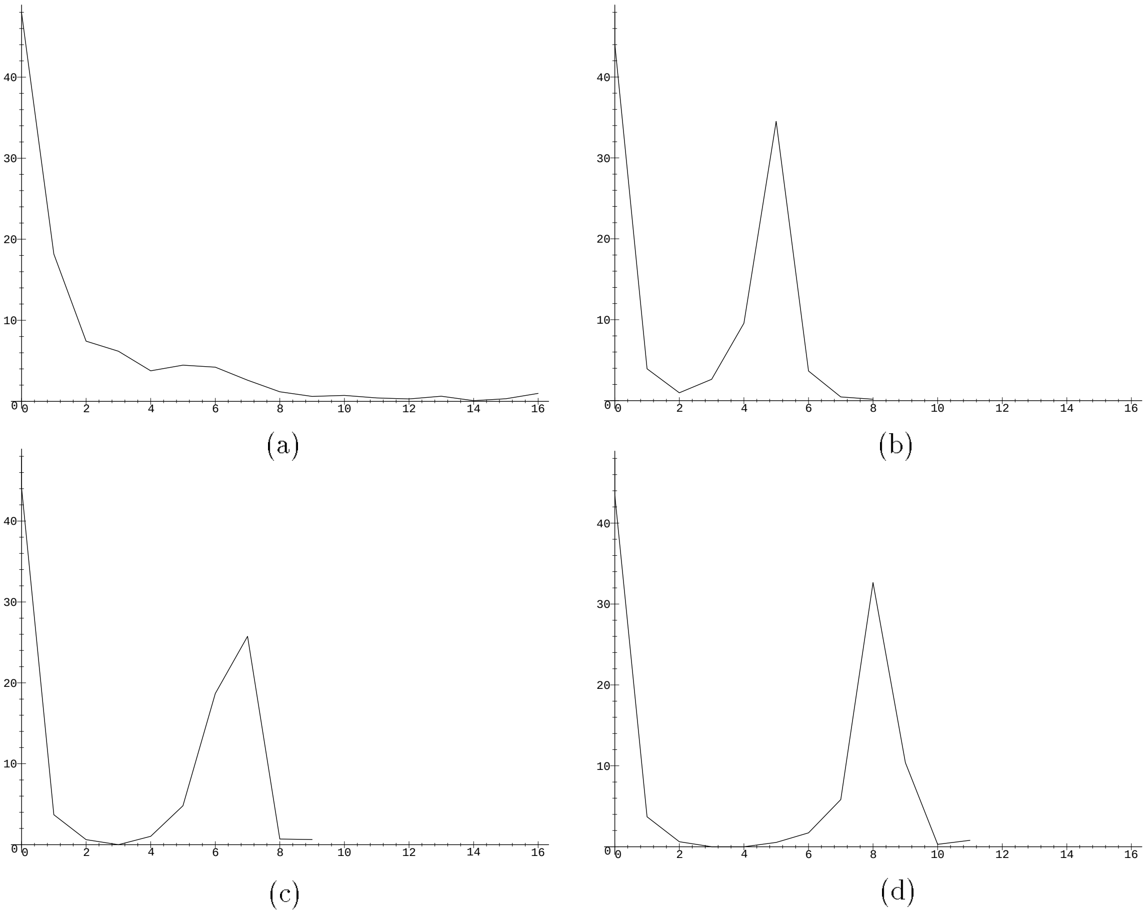

Figure III.18 *Repartition des courbes de Bézier par nombre de subdivisions appliquées : (a) enveloppe convexe topologique ; (b), (c), (d) enveloppe convexe approchée respectivement pour* $\varepsilon = 10^{-1}$, $\varepsilon = 10^{-2}$, $\varepsilon = 10^{-3}$

Le graphe correspondant à l'algorithme de calcul de l'enveloppe convexe topologique nous amène des nouvelles informations, car la concentration des courbes vers les petits nombres de subdivisions n'était pas prévisible d'après le graphe III.17(a). À partir de 9 subdivisions, tous les pourcentages sont inférieurs à 1 (moins de 5% des courbes sont subdivisées 9 fois ou plus) et le graphe est à peu près décroissant. Ceci confirme le fait que les test d'arrêt utilisés sont très sélectifs, les subdivisions s'effectuent de manière très ponctuelle.

Pour conclure cette section, nous remarquons encore une fois que le calcul de l'enveloppe convexe approchée et celui de la structure topologique de l'enveloppe convexe sont des problèmes fondamentalement différents. Nous avons testé si les polygones de contrôle qui fournissent une approximation de l'enveloppe convexe satisfont aussi les critères de la pertinence topologique restreinte, et nous avons trouvé que parmi ceux fournissant une approximation pour $\varepsilon = 10^{-1}$ 55,3% ne le font pas, pour $\varepsilon = 10^{-2}$ il y a 44,7% qui ne vérifient pas ces critères et enfin parmi les polygones de contrôle donnant une approximation de l'enveloppe convexe de la polycourbe à $\varepsilon = 10^{-3}$ près il y a encore 29,41% dont l'enveloppe convexe n'est pas topologiquement pertinente pour celle de la polycourbe.

Dans la dernière section de ce chapitre peuvent être trouvés quelques exemples de polycourbes faisant partie de notre échantillon.

III.8. Exemples

Présentons d'abord le déroulement de l'algorithme de calcul de la structure topologique de l'enveloppe convexe sur quelques exemples de polycourbes de notre échantillon. Nous ne ferons pas une présentation similaire pour l'algorithme de calcul de l'enveloppe convexe approchée, car son intérêt est moindre : toutes les courbes de Bézier identifiées par $\widetilde{\mathcal{I}}_0$ seront subdivisées.

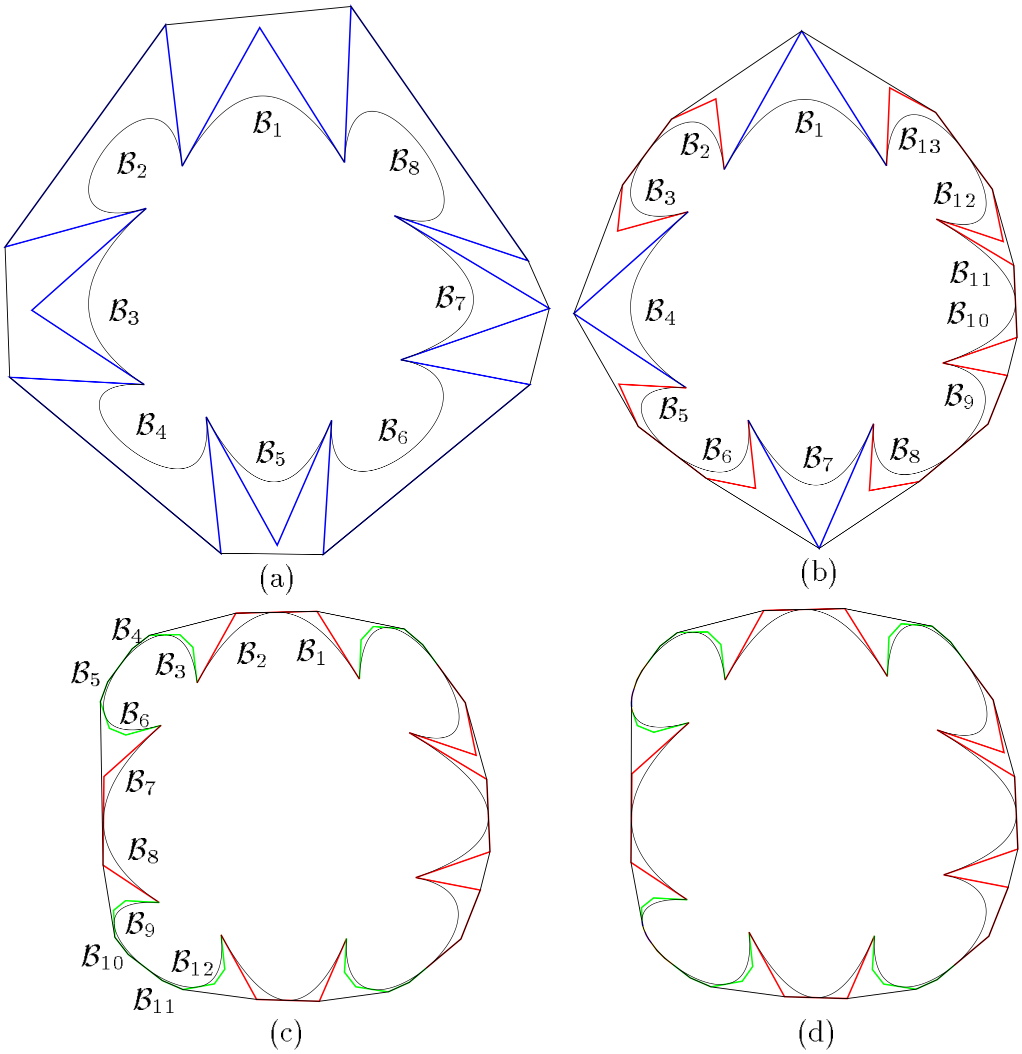

Figure III.19 (a) *aucune des courbes* $\mathcal{B}_2, \mathcal{B}_4, \mathcal{B}_6, \mathcal{B}_7, \mathcal{B}_8$ *ne satisfait aucune des conditions (III.9.i), (III.9.ii), (III.9.iii)* ; (b) *les courbes* $\mathcal{B}_1, \mathcal{B}_4, \mathcal{B}_7$ *ne satisfont pas (III.9.iii), les courbes* $\mathcal{B}_2, \mathcal{B}_5, \mathcal{B}_8$ *ne satisfont pas (III.9.i), les courbes* $\mathcal{B}_3, \mathcal{B}_6, \mathcal{B}_{13}$ *ne satisfont pas (III.9.ii)* ; (c) *aucun point de contrôle de* $\mathcal{B}_7$ *n'est un sommet de* $\mathcal{Q}$*, et par conséquent* $\mathcal{B}_5$ *ne satisfait pas (III.9.ii) et* $\mathcal{B}_{10}$ *ne satisfait pas (III.9.i)* ; (d) *après 6 itérations, l'enveloppe convexe du* *polygone de contrôle est topologiquement pertinente au sens restreint pour celle de la polycourbe (nous n'avons plus représenté les itérations 3,4 et 5)*

L'exemple suivant présente une situation très intéressante. Si les programmes de calcul de l'enveloppe convexe (aussi bien celui pour l'enveloppe convexe approchée que celui pour l'enveloppe convexe topologique) testent la pseudo-simplicité du polygone de contrôle de la polycourbe initiale, donnant un message d'erreur si elle n'est pas satisfaite, ce test serait trop coûteux à faire à chaque itération. Ainsi, les cas comme le suivant peuvent apparaître.

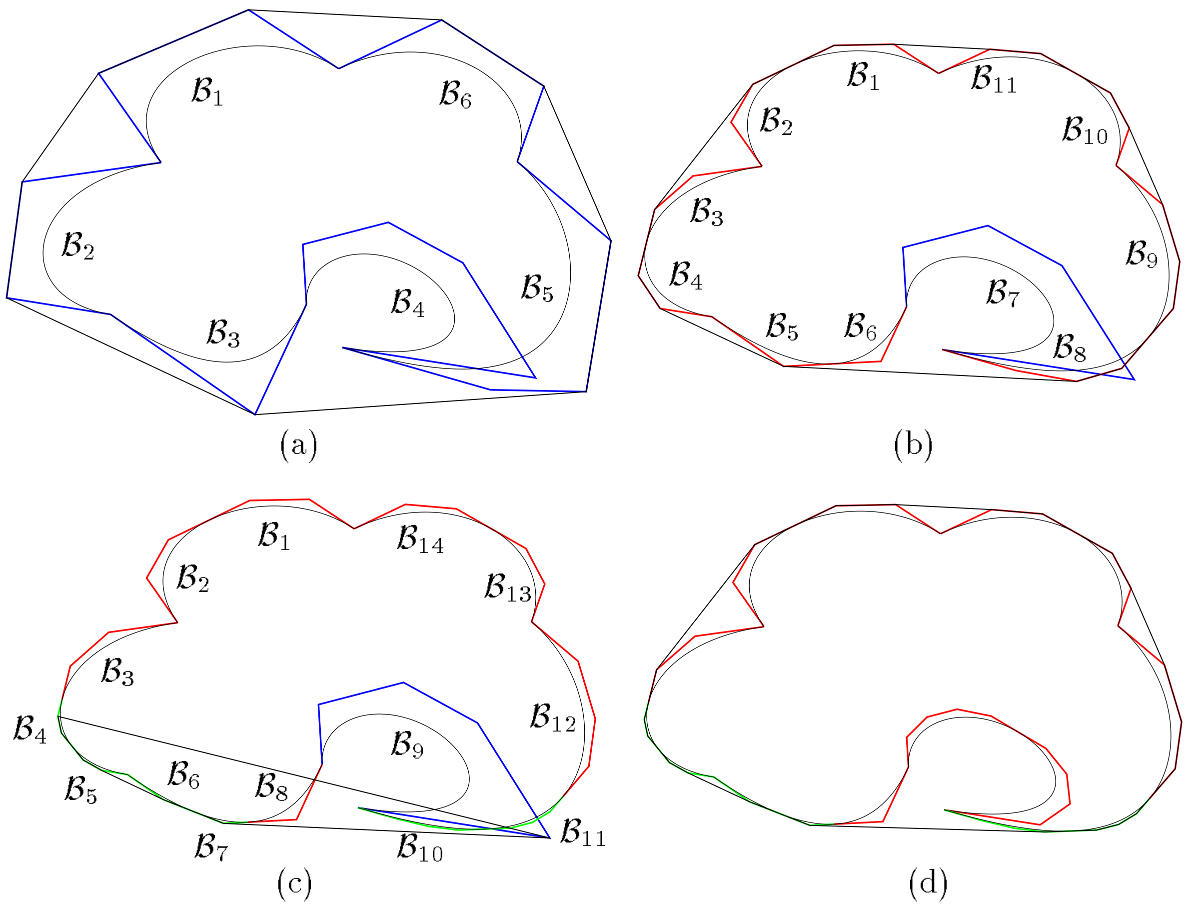

Figure III.20 (a) *les courbes $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_5, \mathcal{B}_6$ ne satisfont au moins une des conditions (III.9.i), (III.9.ii), (III.9.iii) ;* (b) *le sommet $\mathrm{P}_4^{(7)}$ se trouve à gauche de la droite orientée $\mathrm{P}_1^{(5)}\mathrm{P}_2^{(8)}$, donc la ligne polygonale $\mathrm{P}_1^{(5)}\mathrm{P}_0^{(6)}\mathrm{P}_1^{(6)}\mathrm{P}_0^{(7)}\mathrm{P}_1^{(7)} \ldots \mathrm{P}_4^{(7)}\mathrm{P}_0^{(8)}\mathrm{P}_1^{(8)}\mathrm{P}_2^{(8)}$ est identifiée comme une "poche" du polygone $\mathcal{P}$, même si celui-ci n'est pas simple, et les courbes $\mathcal{B}_5$ et $\mathcal{B}_8$ doivent être subdivisées ;* (c) *cette fois, $\mathrm{P}_4^{(9)}$ se trouve à droite de la droite orientée réalisant l'enveloppe convexe droite des polygones $\mathcal{P}_7$ et $\mathcal{P}_{10}$, et comme le polygone de contrôle de la polycourbe est inclus dans le demi-plan gauche délimité par la droite orientée $\mathrm{P}_1^{(7)}\mathrm{P}_4^{(9)}$, l'algorithme de calcul de l'enveloppe convexe du polygone de contrôle trouve un objet complètement erroné. La courbe $\mathcal{B}_9$ ne satisfait pas (III.9.i) et doit être subdivisée ;* (d) *le polygone de contrôle redevient simple, et il le restera jusqu'à ce que les conditions de pertinence topologique* *restreinte soient satisfaites. Au total, six itérations sont nécessaires.*

Nous n'avons pas fait une étude complète de ce type de situation, mais pour les exemples donnés les résultats ont toujours été corrects, même si à une étape intermédiaire des erreurs étaient apparues. La question qui se pose est s'il est possible qu'une courbe de Bézier dont le B-polygone se trouve dans une telle position génératrice de problèmes peut satisfaire les trois conditions (III.9.i), (III.9.ii) et (III.9.iii), ou si elle va toujours être subdivisée jusqu'à la disparition du problème.

Les deux exemples suivants montrent respectivement l'importance des conditions (III.9.i) et (III.9.iii) pour la pertinence topologique restreinte de l'enveloppe convexe du polygone de contrôle pour celle de la polycourbe. La condition (III.9.ii) est la symétrique de (III.9.i).

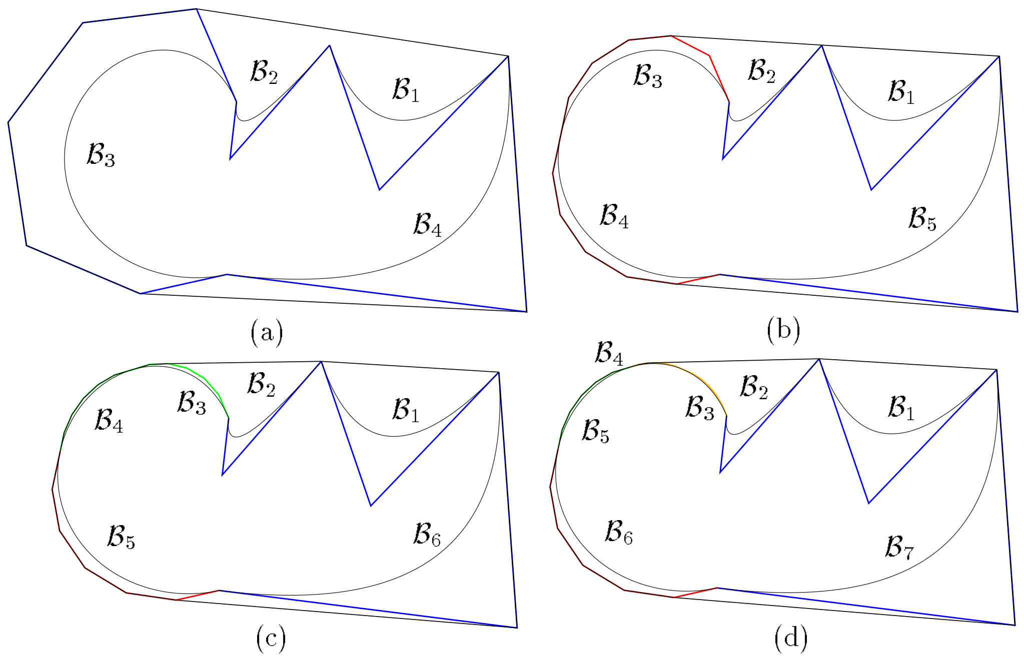

Figure III.21 (a) *$\mathcal{B}_3$ ne satisfait pas (III.9.i) et $\mathcal{E}(\mathcal{P})$ n'est pas topologiquement pertinente pour $\mathcal{E}(\mathcal{B})$;* (b), (c) *$\mathcal{B}_3$ ne satisfait pas (III.9.i), mais il y a pertinence topologique ;* (d) *$\mathcal{B}_4$ satisfait (III.9.i).*

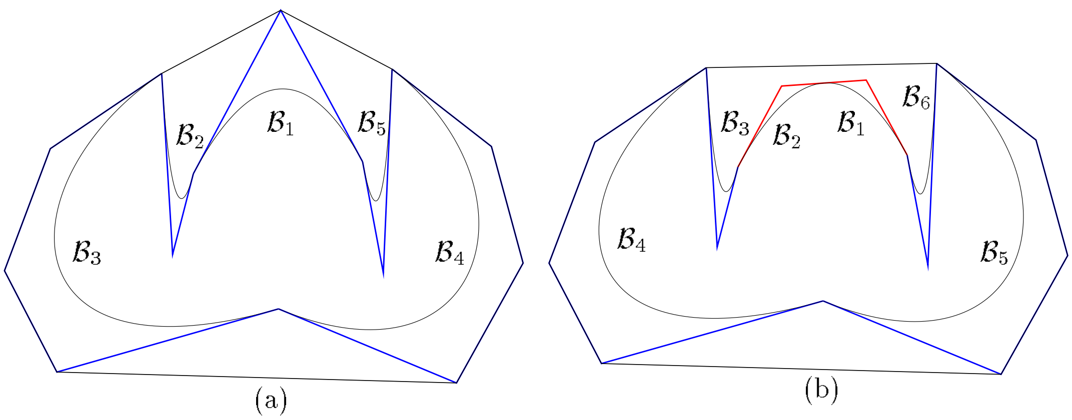

Figure III.22 (a) *les courbes $\mathcal{B}_1, \mathcal{B}_3, \mathcal{B}_4$ satisfont (III.9.i), (III.9.ii), mais $\mathcal{B}_1$ ne satisfait pas (III.9.iii) et l'enveloppe convexe de $\mathcal{P}$ n'est pas topologiquement perti-*

nente pour celle de $\mathcal{B}$; (b) *les trois conditions sont satisfaites et nous avons pertinence topologique restreinte de $\mathcal{E}(\mathcal{P})$ pour $\mathcal{E}(\mathcal{B})$.*