

Courbes de Bézier

Ce cours est une **compilation** :

- Du cours de Modélisation géométrique (IRIT-UPS Toulouse; Equipe Vortex)
 - Cours de Christian Jacquemin (LIMSI- Paris 11)
 - Cours de Marc Daniel (LSIS- Marseille)
- Cours G. Gesquière Master Imagina, DUT Informatique- Arles
 - Cours G. Gesquière, Master Gamagora- Lyon

Un modèle de courbes paramétriques

$$p(u) = \sum_{i=0}^n N_i^d(u) P_i \quad u \in [a, b]$$

avec

$$\sum_{i=0}^n N_i^d(u) = 1 \quad \forall u \in [a, b]$$

- un point de la courbe est une combinaison affine des points de contrôle P_i

Ainsi la position des points de la courbe relativement aux points de contrôle reste invariante par transformation affine.

C'est à dire que pour toute transformation affine Φ , la courbe image $\Phi(p(u))$ a pour points de contrôle les points $\Phi(P_i)$.

Polynôme de Bernstein

- L'idée est de partir du développement binomial :

$$1 = (u + (1-u))^n = \sum_{i=0}^n \binom{n}{i} u^i (1-u)^{n-i}$$

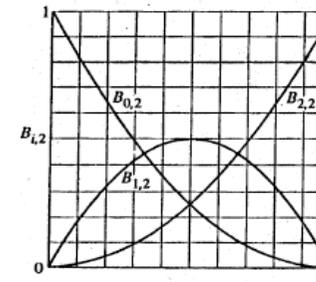
Ainsi, on obtient une somme de $n+1$ polynômes appelés : **polynômes de Bernstein de degré n** :

$$B_i^n(u) = \binom{n}{i} u^i (1-u)^{n-i}, \quad i = 0, \dots, n$$

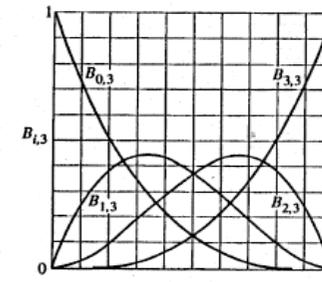
où
$$\binom{n}{i} = \frac{n!}{i!(n-i)!}$$

Graphe des Polynômes de Bernstein

Quelques exemples

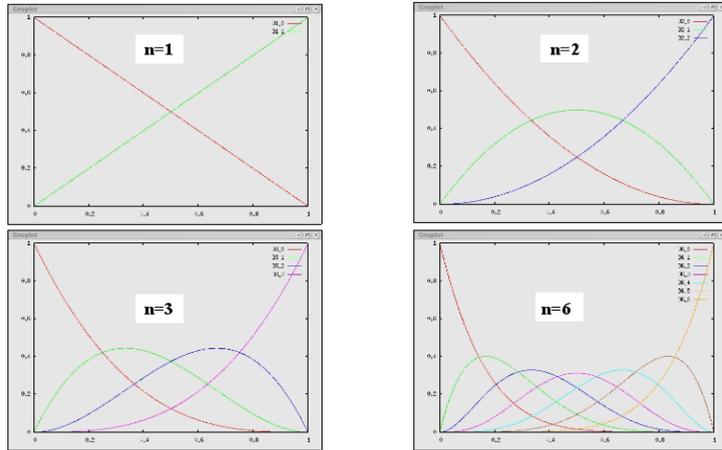


$$\begin{cases} B_{0,2}(t) = t^2 - 2t + 1 \\ B_{1,2}(t) = -2t^2 + 2t \\ B_{2,2}(t) = t^2 \end{cases}$$



$$\begin{cases} B_{0,3}(t) = -t^3 + 3t^2 - 3t + 1 \\ B_{1,3}(t) = 3t^3 - 6t^2 + 3t \\ B_{2,3}(t) = -3t^3 + 3t^2 \\ B_{3,3}(t) = t^3 \end{cases}$$

Graphes des polynômes de Bernstein



Quelques propriétés

- **Propriétés :**
 - pour un degré fixé, ils sont linéairement indépendants,
 - ils sont symétriques : $B_i^n(u) = B_{n-i}^n(1-u)$
 - ils forment une partition de l'unité : $\sum_{i=0}^n B_i^n(u) = 1 \quad \forall u \in \mathbb{R}$
 - ils sont positifs pour u dans [0,1] : $B_i^n(u) > 0 \quad \forall u \in [0,1]$
 - ils satisfont la formule récursive :

$$B_i^{n+1}(u) = u B_{i-1}^n(u) + (1-u) B_i^n(u)$$
 avec : $B_{-1}^n = B_{n+1}^n = 0$ et $B_0^0 = 1$

Calcul des polynômes de Bernstein

- Schéma triangulaire de calcul des polynômes de Bernstein :

$$\begin{array}{ccccccc}
 1 & = & B_0^0 & B_0^1 & B_0^2 & \dots & B_0^n \\
 & & & B_1^1 & B_1^2 & \dots & B_1^n \\
 & & & & B_2^2 & \dots & B_2^n \\
 & & & & & \ddots & \vdots \\
 & & & & & & B_n^n
 \end{array}$$

$\times u$
 $+$
 $\times (1-u)$

- **Exercice :**
 - Calculez les polynômes de Bernstein de degré 4 avec ce schéma.

Courbes de Bézier

- **Courbe de Bézier :**

$$p(u) = \sum_{i=0}^n B_i^n(u) P_i \quad , \quad u \in [0,1]$$
 - les points P_i ($i=0..n$) sont les **n+1 points de contrôle** de la courbe.
 - la courbe est d'**ordre n+1** et son **degré** est **n**.
 - les B_i^n sont les polynômes de Bernstein de degré n. Ils définissent les **fonctions de base** (ou fonction de mélange) de la courbe
- **Le nombre de points de contrôle est directement lié au degré de la courbe : degré n \leftrightarrow n+1 points de contrôle.**
- **Exercice :**
 - Soit la courbe de Bézier contrôlée par les quatre points $P_0(0,0)$, $P_1(5,5)$, $P_2(10,5)$, $P_3(15,0)$.
 - Calculez $p(0)$, $p(1/4)$, $p(1/2)$, $p(3/4)$, $p(1)$ en fonction des P_i , puis faites l'application numérique et tracez la courbe.

```

Tracé d'une courbe de Bézier
> restart;
> xx:=[0,5,10,15]; yy:=[0,5,5,0];

```

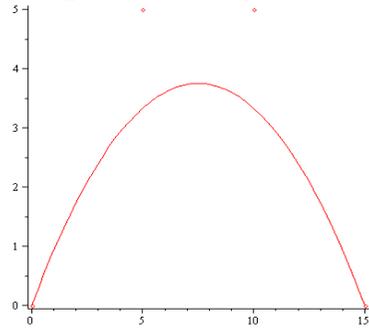
```

> PNTS:=plot([seq([xx[i], yy[i]],i=1..4)],style=point): n:=nops(xx)-1;

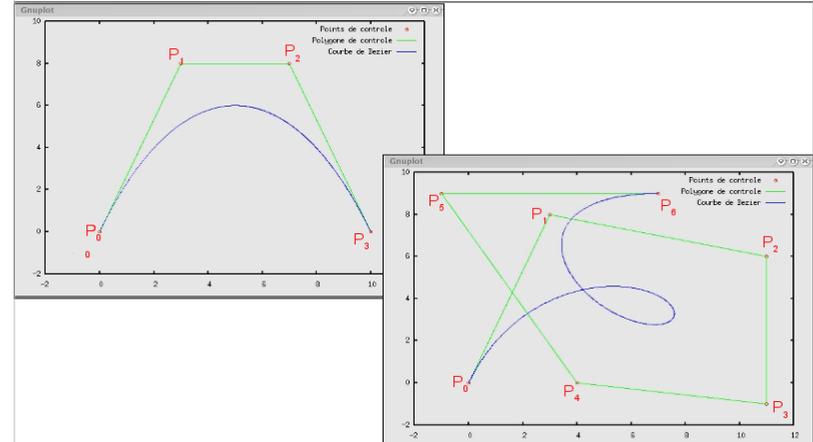
> bez:=x->evalm(sum(binomial(n,i)*x^i*(1-x)^(n-i)*[xx[i+1],yy[i+1]],i=0..n)): PBEZ:=plot([bez(x)[1],bez(x)[2],x=0..1]);

> plots[display]([PNTS, PBEZ], axes=framed);

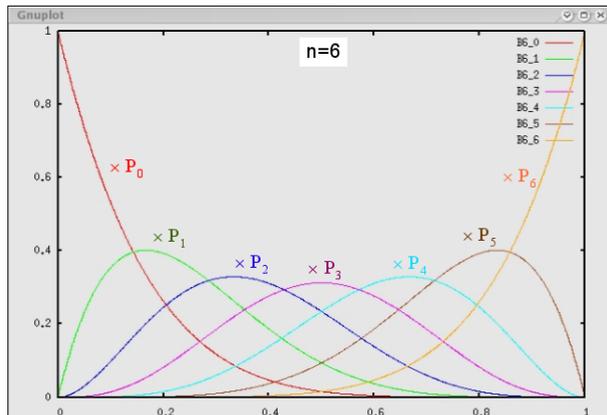
```



Exemples de courbes de Bézier



Portée des fonctions de base



Quelques propriétés

- La symétrie du polynôme de Bernstein implique que :

$$p(u) = \sum_{i=0}^n B_i^n(u) P_i = \sum_{i=0}^n B_i^n(1-u) P_{n-i}$$

- ainsi, la courbe est la même qu'elle soit parcourue de 0 à 1 ou de 1 à 0

- Soit $t \in [a,b]$, $t = a(1-u) + b(u)$, $a \neq b$,

alors :

$$p(t(u)) = p(t) = \sum_{i=0}^n B_i^n(u) P_i$$

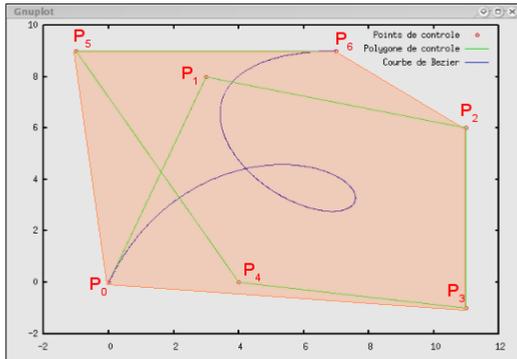
- La courbe de Bézier interpole le premier et le dernier point de contrôle ($u \in [0,1]$) :

$$p(0) = P_0 \quad p(1) = P_n$$

- Elle est tangente au premier et au dernier segment du polygone de contrôle.

Enveloppe convexe

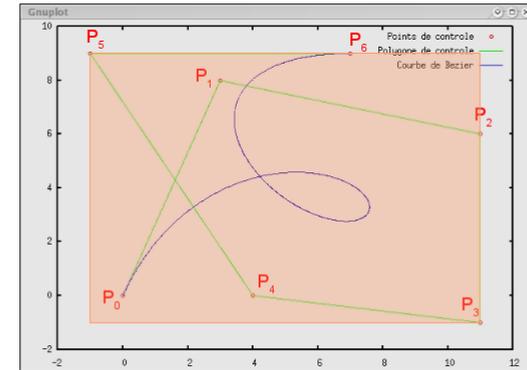
- La courbe est incluse dans l'enveloppe convexe de son polygone de contrôle (car les polynômes de Bernstein sont positifs sur $[0,1]$).



Se rappeler du cas du barycentre d'un ensemble de points (qui sera compris entre ces points).

Boîte englobante

- En prenant individuellement le min et le max de chaque coordonnée des points de contrôle, on obtient une boîte englobante de la courbe qui est parallèle aux axes :



Dérivée d'une courbe de Bézier

• Dérivée :

- Montrez que :
$$\frac{d}{du} B_i^n(u) = n(B_{i-1}^{n-1}(u) - B_i^{n-1}(u))$$

• En déduire que :

$$\frac{dp}{du}(u) = p'(u) = n \sum_{i=0}^{n-1} B_i^{n-1}(u) \Delta P_i, \quad \Delta P_i = P_{i+1} - P_i$$

• A noter :

$$\frac{dp}{dt}(t) = p'(t) = \frac{n}{b-a} \sum_{i=0}^{n-1} B_i^{n-1}(u) \Delta P_i, \quad t \in [a, b], \quad u \in [0, 1]$$

• Exercice :

- Exprimez $p'(0)$, $p'(1/2)$, $p'(1)$ en fonction des P_i pour une courbe de Bézier de degré 3

Algorithme de De Casteljuu

- Cet algorithme s'appuie sur la formule de récurrence suivante :

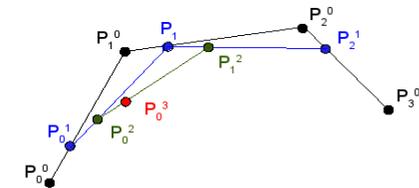
$$p(u) = \sum_{i=0}^n B_i^n(u) P_i^0 = \sum_{i=0}^{n-1} B_i^{n-1}(u) P_i^1 = \dots = \sum_{i=0}^0 B_i^0(u) P_i^n = P_0^n$$

où

$$P_i^{k+1} = (1-u) P_i^k + u P_{i+1}^k$$

Exemple avec $n=3$ et $u=1/4$:

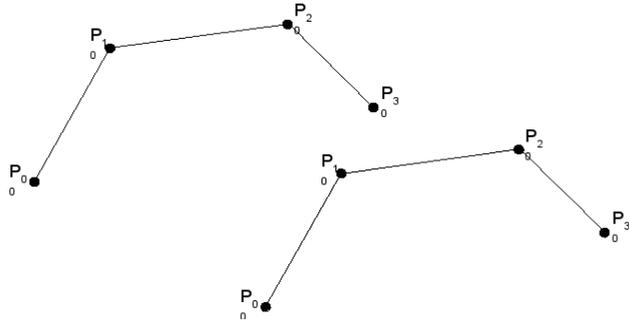
P_0^3 est le point $p(1/4)$



Algorithme de De Casteljau

• Exercice :

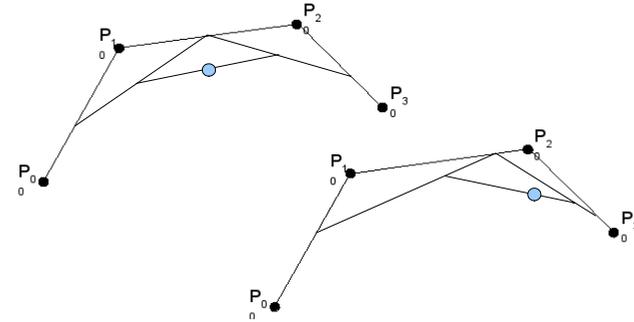
- Appliquez l'algorithme de De Casteljau pour tracer $p(1/2)$ et $p(3/4)$



Algorithme de De Casteljau

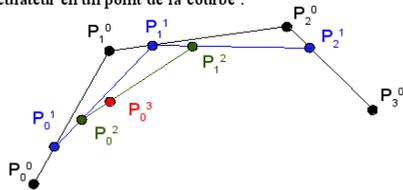
• Exercice :

- Appliquez l'algorithme de De Casteljau pour tracer $p(1/2)$ et $p(3/4)$



Algorithme de De Casteljau

• Tangente et plan osculateur en un point de la courbe :

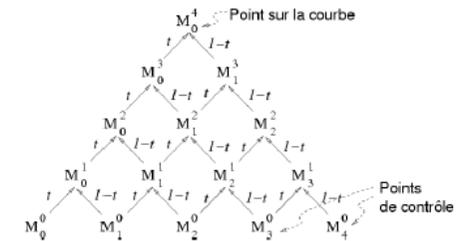


- La tangente en $P_0^3 = p(1/4)$ est portée par le segment $[P_0^2, P_1^2]$
 - En général : $[P_0^{n-1}, P_1^{n-1}]$
- Le plan osculateur en P_0^3 est le plan (P_0^1, P_1^1, P_2^1)
 - En général : $(P_0^{n-3}, P_1^{n-3}, P_2^{n-3})$

Algorithme de De Casteljau sur n points

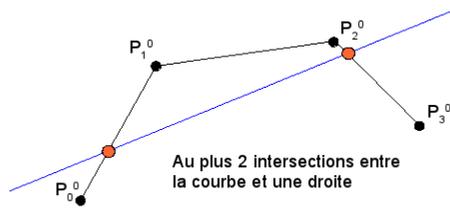
Définition récursive de de Casteljau (suite et fin)

Treillis illustrant le calcul récursif des barycentres dans le cas d'une courbe à 5 points de contrôle.



Propriété de variation

- Une courbe de Bézier ne peut pas avoir plus d'intersections avec une droite que le maximum d'intersection possible entre son polygone de contrôle et une droite quelconque.

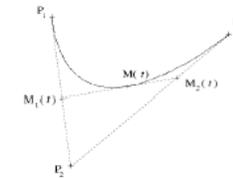


COURBE DE BÉZIER SUR 3 POINTS 1/2

Définition récursive de de Casteljau

Une courbe de Bézier peut se définir comme une **construction récursive de barycentres** dans les rapports $(1 - t)$ et t . Le segment $[M_1(t)M_2(t)]$ est **tangent** à la courbe en $M(t)$.

- Niveau 1 sur (P_1, P_2) : $M_1(t) = (1 - t) P_1 + t P_2$
- Niveau 1 sur (P_2, P_3) : $M_2(t) = (1 - t) P_2 + t P_3$
- Niveau 2 sur (P_1, P_2, P_3) : $M(t) = (1 - t) M_1(t) + t M_2(t) = (1 - t)^2 P_1 + 2 t(1 - t) P_2 + t^2 P_3$



Calcul matriciel avec les polynômes de Bernstein sur 3 points

Les coordonnées d'un point s'obtiennent comme le produit d'une matrice de monomes du paramètre t , une matrice de coefficients et une matrice de points de contrôle P_1, P_2, P_3 .

$$M(t) = (1 - t)^2 P_1 + 2 t(1 - t) P_2 + t^2 P_3$$

$$= B_0^2(t) P_1 + B_1^2(t) P_2 + B_2^2(t) P_3$$

$$M(t) = \begin{bmatrix} B_0^2(t) & B_1^2(t) & B_2^2(t) \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

$$M(t) = \begin{bmatrix} t^2 & t^1 & t^0 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

Calcul matriciel avec les polynômes de Bernstein sur 4 points

Les coordonnées d'un point s'obtiennent comme le produit d'une matrice de monomes du paramètre t , une matrice de coefficients et une matrice de points de contrôle P_1, P_2, P_3, P_4 .

$$M(t) = (1 - t)^3 P_1 + 3 t(1 - t)^2 P_2 + 3 t^2(1 - t) P_3 + t^3 P_4$$

$$= B_0^3(t) P_1 + B_1^3(t) P_2 + B_2^3(t) P_3 + B_3^3(t) P_4$$

$$M(t) = \begin{bmatrix} B_0^3(t) & B_1^3(t) & B_2^3(t) & B_3^3(t) \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}$$

$$M(t) = \begin{bmatrix} t^3 & t^2 & t^1 & t^0 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}$$

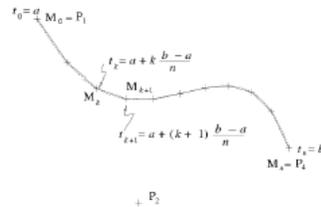
COURBE DE BÉZIER : POLYGONISATION 1/3

Approximation d'une courbe paramétrique par un polygone

Polygonisation d'une courbe de Bézier en n pas avec un paramètre t défini sur $[a,b]$.

Le paramètre t varie de a à b avec un pas de $b-a/n$:

Polygonisation d'une courbe paramétrique (n pas)



30

COURBE DE BÉZIER : POLYGONISATION 2/3

Raffinements

Optimisation

- (1) Calculer le pas de maillage
- (2) calculer toutes les valeurs par additions du pas à la valeur précédente.

Inconvénient: erreurs cumulées.

Maillage adaptatif

Intervalle de maillage **variable** dépendant d'une mesure: longueur du segment, courbure de la courbe.

COURBE DE BÉZIER : POLYGONISATION 3/3

Algorithme de polygonisation d'une courbe paramétrique

Polygonisation d'une courbe de Bézier en n pas avec un paramètre t défini sur $[a,b]$.

```
// initialisation
t = a
P = Bezier( t );
pas = (b - a) / n

// boucle de traçage
Pour i = 1 ; i <= n ; i++
  t' = t + pas;
  P' = Bezier( t' );
  Segment( P , P' );

  t = t';
  P = P';
Fin pour
```

COURBE DE BÉZIER : EXEMPLES D'APPLICATIONS

Dessin interactif de courbes

Dans de nombreuses applications de **modélisation interactive** (image de synthèse, dessin vectoriel, CAO...).

Polices vectorielles

Les **polices vectorielles** postscript sont composées à partir de courbes de Bézier.

Elles sont donc grossissables sans problème d'aliasing.



Caractère tracé à partir
de 4 courbes de Bézier
et 3 segments
(échelles 1/1 et 5/1).

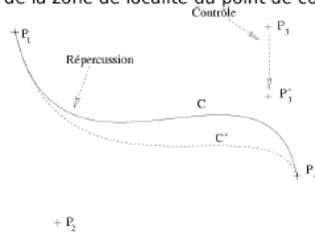
P

★

COURBE DE BÉZIER : PROBLÈME DU CONTRÔLE SUR N POINTS

Non localité du contrôle

Le déplacement d'un point de contrôle sur une courbe de Bézier a des répercussions au-delà de la zone de localité du point de contrôle.



★

FONCTIONS DE PONDÉRATION IDÉALES

Propriétés attendues des fonctions de pondération

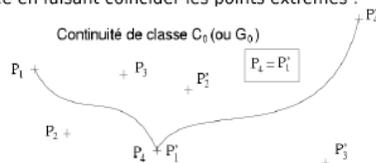
- Interpolation à certains points de contrôle
- Influence locale (non étendue à l'ensemble de la zone).
- Sommation à l'unité pour toute valeur de t
- Bonne continuité

Raccordement de deux courbes

• **Exercice :**

- soient deux courbes de Bézier :
 - $p(u)$ de degré n , $u \in [0, 1]$, points de contrôle P_i
 - $q(v)$ de degré m , $v \in [0, 1]$, points de contrôle Q_j
- Donnez les conditions sur les polygones de contrôle pour que les courbes soient raccordées en $u=1$ et $v=0$ avec :
 - une continuité C^0

Elle est assurée en faisant coïncider les points extrêmes :



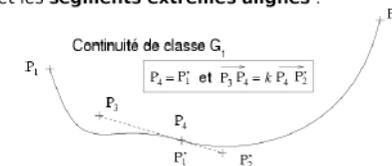
Raccordement de deux courbes

• **Exercice :**

- soient deux courbes de Bézier :
 - $p(u)$ de degré n , $u \in [0, 1]$, points de contrôle P_i
 - $q(v)$ de degré m , $v \in [0, 1]$, points de contrôle Q_j
- Donnez les conditions sur les polygones de contrôle pour que les courbes soient raccordées en $u=1$ et $v=0$ avec :

Continuité de classe G_1

La continuité de classe G_1 est vérifiée ssi les **points extrêmes sont confondus** et les **segments extrêmes alignés** :



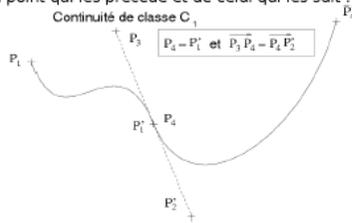
Raccordement de deux courbes

Exercice :

- soient deux courbes de Bézier :
 - $p(u)$ de degré n , $u \in [0,1]$, points de contrôle P_i
 - $q(v)$ de degré m , $v \in [0,1]$, points de contrôle Q_j
- Donnez les conditions sur les polygones de contrôle pour que les courbes soient raccordées en $u=1$ et $v=0$ avec :

Continuité de classe C_1

La continuité de classe C_1 est vérifiée ssi les **points extrêmes sont confondus** et **situés au milieu** du point qui les précède et de celui qui les suit :



★★★

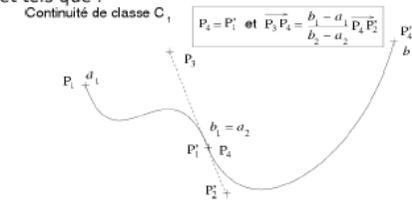
COURBE DE BÉZIER : PROPRIÉTÉS 8/8

Continuité entre deux courbes de Bézier B_1 et B_2 (suite et fin)

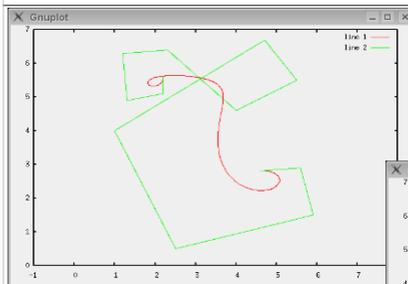
On se place dans le cas où t_1 , le paramètre de B_1 , est défini sur $[a_1, b_1]$ et t_2 , le paramètre de B_2 , sur $[a_2 = b_1, b_2]$.

Continuité de classe C_1

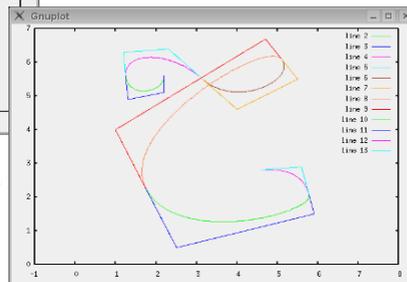
La continuité de classe C_1 est vérifiée ssi les **points extrêmes sont confondus** et tels que :



Exemple de modélisation de courbe complexe



- Une unique courbe de degré 12



- 6 courbes de Bézier de degré 3 (et 2 à la fin) raccordées par continuité C^1

Avantages et inconvénients

- **Avantages :**
 - contrôle intuitif
 - courbe incluse dans le polygone convexe
 - simplicité de mise en oeuvre
- **Désavantages :**
 - support global
 - degré lié au nombre de points de contrôle