

# IV. Design Polynomial

## — Courbes de Bézier

---

### 4.1 COURBES DE BEZIER

Supposons que l'on veuille représenter une courbe sous forme paramétrique par une fonction polynomiale :

$$\begin{aligned}\mathbf{x} : [a, b] \subset \mathbb{R} &\longrightarrow \mathbb{R}^d \\ t &\mapsto \mathbf{x}(t)\end{aligned}$$

Si  $\mathbf{x}$  est exprimée dans la base canonique

$$\mathbf{x}(t) = \sum_{i=0}^n \mathbf{a}_i t^i, \quad \mathbf{a}_i \in \mathbb{R}^d, \quad i = 0, \dots, n,$$

il est très difficile de prévoir l'allure de la courbe  $(\mathcal{C}) = \mathbf{x}([a, b])$  en fonction de la position des points  $\mathbf{a}_0, \dots, \mathbf{a}_n$ .

De même si l'on modifie un des points  $\mathbf{a}_i$ , il est pratiquement impossible de prévoir la modification de la courbe qui va en résulter.

Par contre si on change de base, par exemple si l'on prend la *base de Bernstein* :

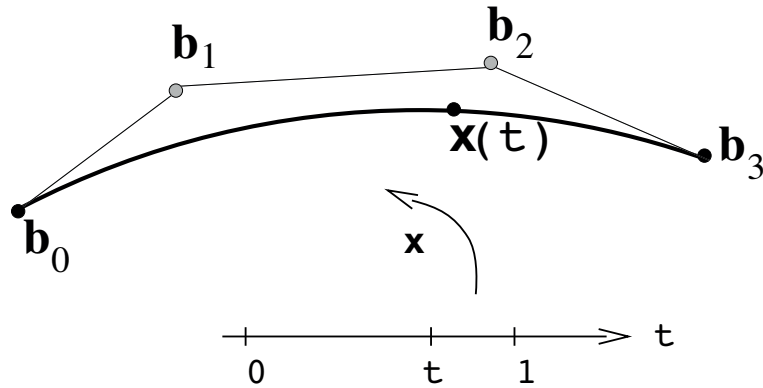
$$B_i^n(t) = \binom{n}{i} (1-t)^{n-i} t^i, \quad \text{où} \quad \binom{n}{i} = \frac{n!}{(n-i)!i!};$$

La courbe

$$\mathbf{x}(t) = \sum_{i=0}^n \mathbf{b}_i B_i^n(t) \quad t \in [0, 1]$$

s'appelle *courbe de Bézier*. Les points  $\mathbf{b}_i$ ,  $i = 0, \dots, n$  s'appellent *points de Bézier* (ou *points de contrôle*) et forment le *polygone de contrôle*.

**Exemple :**  $n = 3$ ,  $d = 2$



et le polygone défini par les points  $\mathbf{b}_i$  donne une bonne idée intuitive de la courbe  $(\mathcal{C}) = \mathbf{x}([0, 1])$ .

Pour un intervalle  $[a, b]$  quelconque, il faut simplement faire un changement de variable  $u(t) = \frac{t-a}{b-a}$  pour revenir à la définition précédente.

## 4.2 PROPRIETES DES POLYNOMES DE BERNSTEIN

Sans restriction de généralité, supposons  $t \in [0, 1]$ .

**(1) Partition de l'unité :**  $\sum_{i=0}^n B_i^n(t) \equiv 1, \forall t \in [0, 1]$ .

On obtient ce résultat par la formule du binôme :

$$\begin{aligned} 1 &\equiv [(1-t) + t]^n \equiv (1-t)^n + \binom{n}{1}(1-t)^{n-1}t + \dots + \binom{n}{i}(1-t)^{n-i}t^i + \dots + t^n \\ &\equiv \sum_{i=0}^n B_i^n(t). \end{aligned}$$

**(2) Positivité :**  $B_i^n(t) \geq 0, \forall t \in [0, 1], 0 \leq i \leq n$ .

**(3) Symétrie :**  $B_i^n(t) = B_{n-i}^n(1-t)$

$$\begin{aligned} B_i^n(t) &= \binom{n}{i}(1-t)^{n-i}t^i = \binom{n}{n-i}(1-(1-t))^{n-(n-i)}(1-t)^{n-i} \\ &= B_{n-i}^n(1-t). \end{aligned}$$

(4) Formule de récurrence :

$$B_i^n(t) = \begin{cases} (1-t)B_i^{n-1}(t) & i = 0 \\ (1-t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t) & i = 1, \dots, n-1 \\ tB_{i-1}^{n-1}(t) & i = n \end{cases}$$

et  $B_0^0(t) \equiv 1$ ,  $B_i^n(t) \equiv 0$  si  $i \notin \{0, \dots, n\}$ .

On sait que pour  $i = 1, \dots, n-1$ , on a

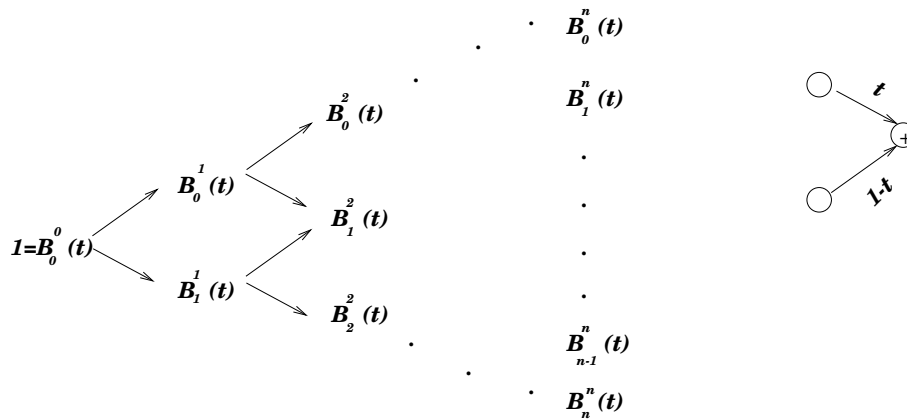
$$\binom{n}{i} = \binom{n-1}{i} + \binom{n-1}{i-1}.$$

C'est une formule classique de combinatoire :

$$\begin{aligned} \binom{n-1}{i} + \binom{n-1}{i-1} &= \frac{(n-1)!}{(n-1-i)!i!} + \frac{(n-1)!}{(n-i)!(i-1)!} \\ &= \frac{(n-1)!(n-i)}{(n-i)!i!} + \frac{(n-1)!i}{(n-i)!i!} \\ &= \frac{(n-1)!(n)}{(n-i)!i!} = \binom{n}{i}. \end{aligned}$$

$$\begin{aligned} B_i^n(t) &= \binom{n}{i} t^i (1-t)^{n-i} \\ &= \binom{n-1}{i} t^i (1-t)^{n-i} + \binom{n-1}{i-1} t^i (1-t)^{n-i} \\ &= (1-t) \left[ \binom{n-1}{i} t^i (1-t)^{n-1-i} \right] + t \left[ \binom{n-1}{i-1} t^{i-1} (1-t)^{n-i} \right] \\ &= (1-t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t). \end{aligned}$$

La formule de récurrence permet de calculer les  $B_i^n(t)$  avec le *tableau triangulaire*, dit *tableau d'Aitken/Mansfield* suivant :



Les  $B_i^n(t)$  peuvent ainsi être construits par interpolation linéaire répétée.

**(5) le maximum :**  $\max_{t \in [0,1]} B_i^n(t) = \frac{n}{i}$ .

**(6) Valeurs aux extrémités :**  $B_i^n(0) = \delta_{i,0}$ ,  $B_i^n(1) = \delta_{i,n}$ .

**(7) Base polynomiale :**  $\{B_i^n, \quad i = 0, \dots, n\}$  constitue une base pour  $\mathcal{P}_n$ .

$B_i^n(t)$  est un polynôme de degré  $n$  a une racine  $i^{\text{ème}}$  en  $t = 0$  et une racine  $(n - i)^{\text{ème}}$  en  $t = 1$ , donc

$$\begin{cases} (B_i^n)^{(j)}(0) = 0, \text{ pour } j = 0, \dots, i - 1 \\ (B_i^n)^{(i)} \neq 0. \end{cases}$$

Supposons  $f(t) = \sum_{i=0}^n \alpha_i B_i^n(t) = 0$ , on a

$$\begin{aligned} f(0) &= \alpha_0 B_0^n(0) + \alpha_1 B_1^n(0) + \dots + \alpha_n B_n^n(0) \\ &= \alpha_0 \underbrace{B_0^n(0)}_{\neq 0} = 0 \implies \alpha_0 = 0 \end{aligned}$$

$$\begin{aligned} f'(0) &= \alpha_1 (B_1^n)'(0) + \dots + \alpha_n (B_n^n)'(0) \\ \alpha_1 (B_1^n)'(0) + 0 + \dots + 0 &= 0 \implies \alpha_1 = 0 \end{aligned}$$

$\vdots$

$$\begin{aligned} f^{(r)}(0) &= \alpha_r (B_r^n)^{(r)}(0) + \dots + \alpha_n (B_n^n)^{(r)}(0) \\ \alpha_r (B_r^n)^{(r)}(0) + 0 + \dots + 0 &= 0 \implies \alpha_r = 0 \end{aligned}$$

**(8) Dérivée :**

- le cas où  $u \in [0, 1]$  :

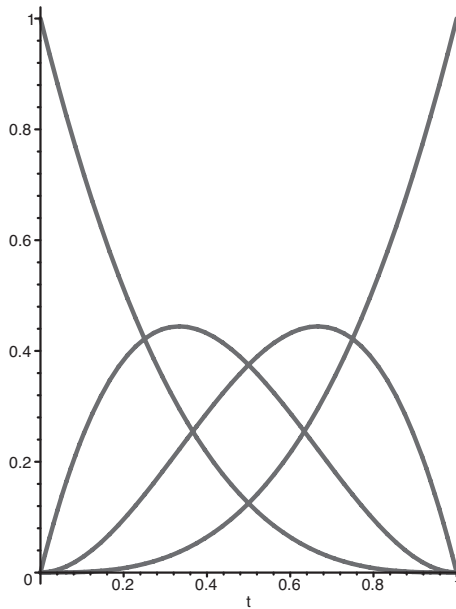
$$(B_i^n)'(t) = \begin{cases} -n B_i^{n-1}(t) & i = 0 \\ n(B_{i-1}^{n-1}(t) - B_i^{n-1}(t)) & i = 1, \dots, n-1 \\ n B_{i-1}^{n-1}(t) & i = n \end{cases}$$

$$\begin{aligned}
(B_i^n)'(t) &= i \binom{n}{i} t^{i-1} (1-t)^{n-i} - (n-i) \binom{n}{i} t^i (1-t)^{n-1-i} \\
&= \frac{i \cdot n!}{i!(n-i)!} t^{i-1} (1-t)^{n-i} - \frac{(n-i) \cdot n!}{i!(n-i)!} t^i (1-t)^{n-1-i} \\
&= n \frac{(n-1)!}{(i-1)!(n-i)!} t^{i-1} (1-t)^{n-1-(i-1)} - n \frac{(n-1)!}{i!(n-1-i)!} t^i (1-t)^{n-1-i} \\
&= n(B_{i-1}^{n-1}(t) - B_i^{n-1}(t)).
\end{aligned}$$

- le cas où  $t \in [a, b]$  :

$$(B_i^n)'(\frac{b-u}{b-a}) = \begin{cases} -\frac{n}{b-a} B_i^{n-1}(\frac{b-u}{b-a}) & i = 0 \\ \frac{n}{b-a} (B_{i-1}^{n-1}(t) - B_i^{n-1}(\frac{b-u}{b-a})) & i = 1, \dots, n-1 \\ \frac{n}{b-a} B_{i-1}^{n-1}(\frac{b-u}{b-a}) & i = n \end{cases}$$

**Exemple : Polynômes de Bernstein cubiques**



$$\begin{aligned}
B_0^3 &= (1-t)^3 \\
B_1^3 &= 3(1-t)^2 t \\
B_2^3 &= 3(1-t) t^2 \\
B_3^3 &= t^3
\end{aligned}$$

### 4.3 ALGORITHME DE DE CASTELJAU (1959)

Historiquement, c'est avec cet algorithme que les travaux de M. de Casteljau commençaient en 1959 chez Citron. Ils étaient publiés comme des rapports techniques, tenus très au secret par Citron.

Ces travaux restaient inconnus jusqu'en 1975 quand W. Böhm en a pris connaissance et les a rendu public.

Au début des années 60, Pierre Bézier commenait ses travaux le système UNISURF chez Renault et les publiait, d'où le fait que le nom de Bézier a été donné à ces courbes et surfaces de forme libre.

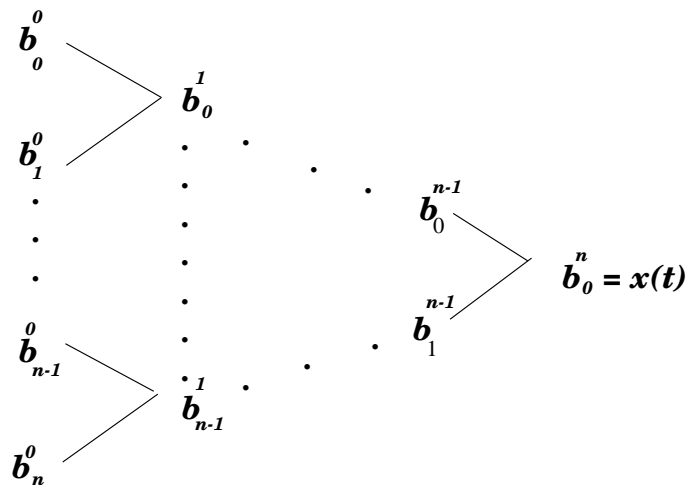
**L'algorithme de de Casteljau** sert à évaluer  $x(t)$  en un paramètre  $t$  donné avec les points auxiliaires de la manière suivante :

$\begin{aligned} b_i^k &= (1-t)b_i^{k-1} + tb_{i+1}^{k-1} \\ b_i^0 &= b_i, \quad i = 0, \dots, n \end{aligned}$	$k = 1, \dots, n, \quad i = 0, \dots, n-k.$
---	---

Il est facile à démontrer que  $x(t) = b_0^n$ .

L'algorithme de de Casteljau est un algorithme triangulaire à  $n$  niveau

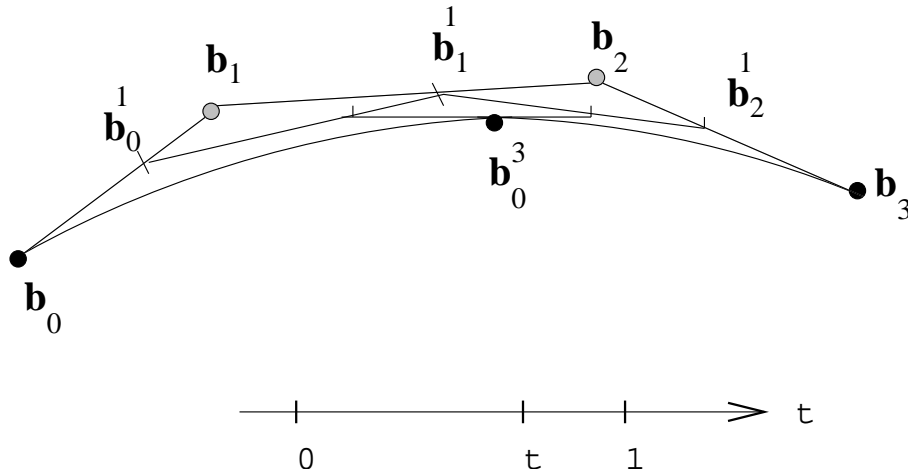
**Schéma :**



Démonstration :

$$\begin{aligned}
x(t) &= \mathbf{b}_0 B_0^n(t) + \mathbf{b}_1 B_1^n(t) + \dots + \mathbf{b}_n B_n^n(t), \quad t \in [0, 1] \\
&\quad (\text{Aitken/Mansfield ont exprimé } B_i^n \text{ en fonction des } B_j^{n-1}) \\
&= \mathbf{b}_0(1-t)B_0^{n-1}(t) + \mathbf{b}_1((1-t)B_1^{n-1}(t) + tB_0^{n-1}(t)) + \dots + \mathbf{b}_n t B_{n-1}^{n-1}(t) \\
&\quad (\text{trier par rapport à } B_i^{n-1}) \\
&= \underbrace{[(1-t)\mathbf{b}_0 + t\mathbf{b}_1]}_{=:\mathbf{b}_0^1} B_0^{n-1}(t) + \dots + \underbrace{[(1-t)\mathbf{b}_{n-1} + t\mathbf{b}_n]}_{=:\mathbf{b}_{n-1}^1} B_{n-1}^{n-1}(t) \\
&\quad \vdots \\
&\quad (\text{répéter ce passage jusqu'à}) \\
&= \underbrace{[(1-t)\mathbf{b}_0^{n-1} + t\mathbf{b}_1^{n-1}]}_{=:\mathbf{b}_0^n} \underbrace{B_0^0(t)}_{\equiv 1} \\
&= \mathbf{b}_0^n
\end{aligned}$$

Graphiquement :



## 4.4 PROPRIETES GEOMETRIQUES DES COURBES DE BEZIER

(a) **Propriété d'enveloppe convexe :**

pour  $t \in [0, 1]$ , la courbe  $\mathcal{C}$  se trouve à l'intérieur de l'enveloppe convexe (intersection de tous les ensembles convexes contenant  $\{\mathbf{b}_i\}$ ) des  $\mathbf{b}_i$ .

C'est une conclusion directe de la positivité et de la partition de l'unité des polynômes de Bernstein ((1) et (2)).

Chaque point intermédiaire  $\mathbf{b}_i^r$  dans l'algorithme de de Casteljau est une combinaison barycentrique convexe des points précédents  $\mathbf{b}_j^{r-1}$ . cela implique par exemple que chaque polygone de contrôle plan génère une courbe plane.

L'importance de cette propriété se trouve dans ce qu'on appelle "*test de collision*" :

Si 2 courbes de Bézier représentent le chemin d'un bras de robot, peut-il y avoir une collision ?

Au lieu de calculer les intersections possibles, on fait le test des "bounding boxes" ou des boîtes englobantes "minmax boxes". (Arêtes parallèles aux axes du système de coordonnées).

C'est un test très rapide s'il y a des milliers d'orbites à tester.

### (b) Invariance affine

Les applications affines sont incluses dans tous les systèmes de CAO : les objets doivent être tournés, repositionnés, chargés de taille, etc.

Les courbes de Bézier sont affinement invariantes, ce qui implique que les deux procédures suivantes mènent au même résultat :

1. calculer d'abord le point de la courbe  $\mathbf{x}(\hat{t})$  et puis appliquer la transformation affine,
2. appliquer d'abord la transformation affine aux points de contrôle et puis évaluer  $\mathbf{x}(\hat{t})$ .

— dessin —

L'invariance affine est une conséquence directe de l'algorithme de de Casteljau qui consiste en la répétition d'interpolations linéaires entre deux points (c.à.d. d'applications affines). ces opérations elles mêmes sont affinement invariantes, donc leur composition l'est aussi

$$f \text{ affine} \iff f(\alpha \mathbf{a} + (1 - \alpha) \mathbf{b}) = \alpha f(\mathbf{a}) + (1 - \alpha) f(\mathbf{b})$$

Par contre, les courbes de Bézier ne sont pas invariantes par projection. Les applications projectives sont par exemple utilisées en infographie pour le rendu réaliste d'objets 3D  $\implies$  application perspective.

**Une transformation affine** est une application  $T$  qui applique  $\mathbb{E}^3$  (espace affine euclidien tridimensionnel) sur lui même et qui laisse les combinaisons barycentriques invariantes.

### Combinaisons barycentriques (affine)

$$\mathbf{b} = \sum_{i=0}^n \alpha_i \mathbf{b}_i, \quad \mathbf{b}_i \in \mathbb{E}^3, \quad \alpha_0 + \dots + \alpha_n = 1$$



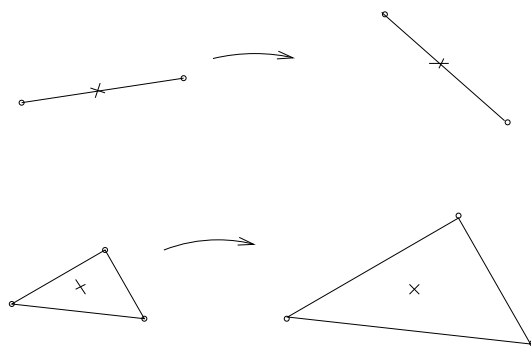
C'est une somme pondérée de points où la somme des poids est égale à 1, c'est l'opération fondamentale pour les points.

Si  $\mathbf{x} = \sum_{i=0}^n \alpha_i \mathbf{a}_i$ ,  $\mathbf{x}, \mathbf{b}_i \in \mathbb{E}^3$  et si  $T$  est une transformation affine, alors également

$$T\mathbf{x} = \sum_{i=0}^n \alpha_i T\mathbf{a}_i; \quad T\mathbf{x}, T\mathbf{a}_i \in \mathbb{E}^3.$$

Par exemple, au point milieu d'un segment de droite correspondra le milieu de l'image affine de ce segment de ligne droite.

A l'isobarycentre de  $n$  points correspond l'isobarycentre des points images.



Soit  $\mathbf{x} \in \mathbb{E}^3$ , une transformation affine est donnée par

$$T\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{V} \quad \begin{array}{l} \mathbf{A} \text{ matrice } 3 \times 3 \\ \mathbf{V} \text{ point de } \mathbb{E}^3. \end{array}$$

Vérification : ( $T$  préserve les combinaisons barycentriques ?)

$$\begin{aligned} T\left(\sum \alpha_i \mathbf{a}_i\right) &= \mathbf{A}\left(\sum \alpha_i \mathbf{a}_i\right) + \mathbf{V} \\ &= \sum \alpha_i \mathbf{A}\mathbf{a}_i + \sum \alpha_i \mathbf{V} & (\sum \alpha_i = 1) \\ &= \sum \alpha_i (\mathbf{A}\mathbf{a}_i + \mathbf{V}) \\ &= \sum \alpha_i T\mathbf{a}_i. \end{aligned}$$

Exemples de transformations affines :

- L'identité ( $\mathbf{V} = \mathbf{o}$ ,  $\mathbf{A} = \mathbf{I}$ ).
- La translation ( $\mathbf{A} = \mathbf{I}$ ).
- L'homothétie ( $\mathbf{V} = \mathbf{o}$ ,  $\mathbf{A}$  diagonale).
- La rotation ( $\mathbf{V} = \mathbf{o}$ , ex. rotation autour de l'axes des  $z$ ) :

$$\mathbf{A} = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Le cisaillement

**Transformations euclidiennes  $\subset$  transformations affines**

$\mathbf{A}^T \mathbf{A} = \mathbf{I}$  ( $\mathbf{A}$  orthogonale).

Elles laissent les longueurs et les mesures d'angles inchangées (ex. rotation, translation).

Les transformation affines peuvent être combinées.

Une transformation compliquée peut être décomposée en une séquence de transformations plus simples.

**Théorème :** Chaque transformation affine peut être composée de translations, de rotations, de cisaillements et d'homothéties.

$\text{Rang}(\mathbf{A}) = 3$  : transforme un objet tridimensionnel en un objet tridimensionnel

$\text{Rang}(\mathbf{A}) = 2$  :  $T$  est une projection parallèle sur un plan

$\text{Rang}(\mathbf{A}) = 1$  :  $T$  est une projection parallèle sur une ligne droite.

$T : \mathbb{E}^2 \longrightarrow \mathbb{E}^2$  est uniquement déterminée par un triangle et son image.

$\implies$  2 triangles quelconques déterminent une transformation du plan dans lui même.

$T : \mathbb{E}^3 \longrightarrow \mathbb{E}^3$  est uniquement déterminée par un tétraèdre et son image.

L'interpolation linéaire :  $\mathbf{x} = \mathbf{x}(t) = (1-t)\mathbf{a} + t\mathbf{b}$ ;  $t \in \mathbb{R}$  est affinement invariante (car c'est une combinaison barycentrique).

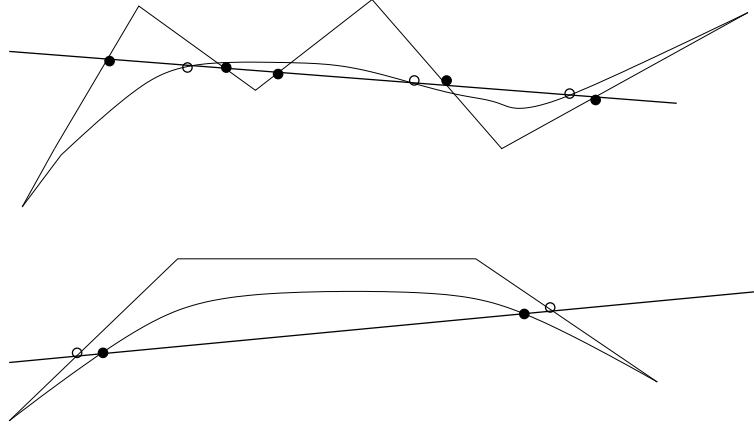
**Démonstration :** soit  $T : \mathbb{E}^3 \longrightarrow \mathbb{E}^3$  une transformation affine, alors :

$$\begin{aligned} T\mathbf{x} &= (1-t)T\mathbf{a} + tT\mathbf{b} \\ &= \mathbf{A}((1-t)\mathbf{a} + t\mathbf{b}) + \mathbf{V} \\ &= (1-t)\mathbf{A}\mathbf{a} + t\mathbf{A}\mathbf{b} + (1-t)\mathbf{V} + t\mathbf{V} \\ &= (1-t)(\mathbf{A}\mathbf{a} + \mathbf{V}) + t(\mathbf{A}\mathbf{b} + \mathbf{V}) \\ &= (1-t)T\mathbf{a} + tT\mathbf{b} \end{aligned}$$

**(c) Diminution de la variation** (Variation Diminishing)

Soit  $\mathcal{C}$  une courbe de Bézier et  $\mathcal{P}$  sont polygone de contrôle. Si l'on dénote  $cross\mathcal{C}$  ou  $cross\mathcal{P}$  respectivement le nombre d'intersection de  $\mathcal{C}$  ou  $\mathcal{P}$  avec un plan quelconque, alors

$$cross\mathcal{C} \leq cross\mathcal{P}.$$



aucune droite n'intersecte  $\mathcal{C}$  plus souvent que  $\mathcal{P}$ .

Cela implique par exemple qu'une polygone convexe génère une courbe convexe.

Attention l'inverse n'est pas vraie.

Démonstration: voir [Farin1992], ou [Schoenberg, Greville: On spline functions, 1976].

Les deux propriétés de l'EC et de la VDP ensemble permettent de bien prédire la forme de la courbe.

La propriété suivante améliore encore la prédiction de la forme de la courbe à partir de son polygone de contrôle :

**(d) Interpolation des extrémités**

$$\mathbf{x}(0) = \mathbf{b}_0$$

$$\mathbf{x}(1) = \mathbf{b}_n$$

la courbe passe par  $\mathbf{b}_0$  et  $\mathbf{b}_n$ . C'est une conséquence directe de la propriété (6) des polynômes de Bernstein :

$$B_i^n(0) = \delta_{i,0}$$

$$B_i^n(1) = \delta_{i,n}$$

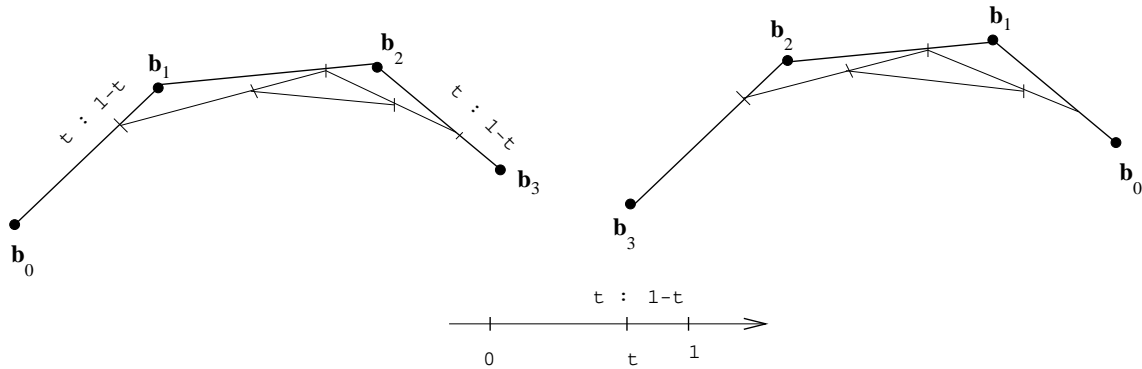
En phase de dessin, c'est une propriété importante de pouvoir contrôler directement les extrémités d'une courbe.

**(e) Symétrie :**

Les polynômes de Bernstein sont symétrique par rapport à  $t$  et  $1 - t$ , donc

$$\sum_{i=0}^n b_i B_i^n(t) = \sum_{i=0}^n b_{n-i} B_i^n(1-t)$$

ce qui change entre les deux courbes c'est leur *sens de parcours*.



**(f) Contrôle pseudo-local :**

Le polynôme de Bernstein n'a qu'un maximum, atteint en  $t = \frac{i}{n}$ .

Ceci a des applications en dessin : si nous déplaçant uniquement un des sommets du polygone de contrôle, par exemple  $b_i$ , alors la courbe est principalement affectée par ce changement dans la région autour des points où le paramètre prend la valeur  $\frac{i}{n}$ .

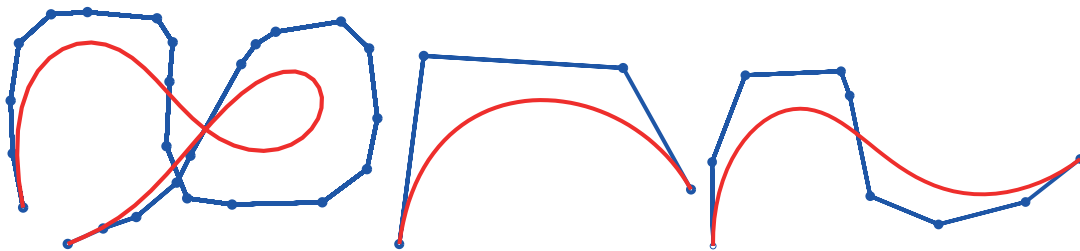
Ceci rend l'effet du changement raisonnablement prévisible, bien que le changement affecte la courbe *globalement*.

**(g) Invariance par transformations affines du paramètre :**

Algébriquement  $\sum_{i=0}^n b_i B_i^n(t) = \sum_{i=0}^n b_i B_i^n\left(\frac{u-a}{b-a}\right)$ ,  $t \in [0, 1]$  et  $u \in [a, b]$ . (L'algorithme de de Casteljau est "aveugle" vis à vis l'intervalle du paramètre. La transition de l'intervalle  $[0, 1]$  à  $[a, b]$  est une carte affine).

**(h) Dessiner avec les courbes de Bézier :**

la courbe 'imite' le parcours du polygone de contrôle.



## 4.5 DERIVATION D'UNE COURBE DE BEZIER

Nous avons vu que  $\frac{d}{dt}B_i^n(t) = n(B_{i-1}^{n-1}(t) - B_i^{n-1}(t))$ , nous pouvons donc déterminer la dérivée d'une courbe de Bézier  $\mathbf{x}(t)$  :

$$\begin{aligned}\frac{d}{dt}\mathbf{x}(t) &= n \sum_{i=0}^n \mathbf{b}_i (B_{i-1}^{n-1}(t) - B_i^{n-1}(t)) \\ &= n \sum_{i=1}^n \mathbf{b}_i B_{i-1}^{n-1}(t) - n \sum_{i=0}^{n-1} \mathbf{b}_i B_i^{n-1}(t).\end{aligned}$$

Par transformation d'indice dans la première somme, on obtient

$$\frac{d}{dt}\mathbf{x}(t) = n \sum_{i=0}^n (\mathbf{b}_{i+1} - \mathbf{b}_i) B_i^{n-1}(t).$$

On peut simplifier la formule en introduisant l'opérateur de différence avant  $\Delta$  :

$$\Delta \mathbf{b}_i = \mathbf{b}_{i+1} - \mathbf{b}_i,$$

d'où l'expression suivante pour la dérivée d'une courbe de Bézier :

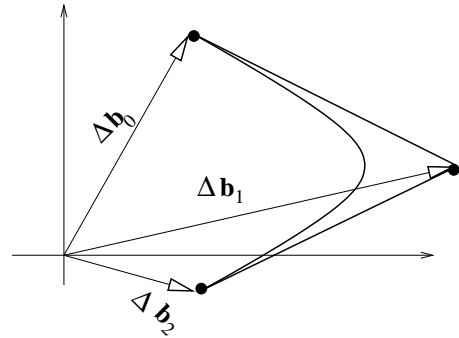
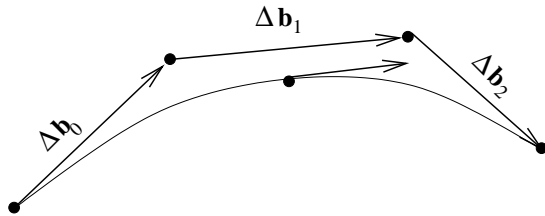
$$\frac{d}{dt}\mathbf{x}(t) = n \sum_{i=0}^n \Delta \mathbf{b}_i B_i^{n-1}(t), \quad \Delta \mathbf{b}_i \in \mathbb{R}^d.$$

La dérivée d'une courbe de Bézier est donc une autre courbe de Bézier, obtenue par application de l'opérateur différence sur le polygone de contrôle originale.

Cependant, cette courbe de Bézier dérivée n'est plus du tout dans  $\mathbb{E}^d$ !

Ces coefficients sont des différences de points, c.à.d. des vecteurs éléments de  $\mathbb{R}^d$ .

Afin de visualiser cette courbe dérivée et son polygone dans  $\mathbb{E}^d$ , nous pouvons construire un polygone dans  $\mathbb{E}^d$  qui consiste des points  $\mathbf{a} + \Delta \mathbf{b}_0, \dots, \mathbf{a} + \Delta \mathbf{b}_{n-1}$ . Ici  $\mathbf{a}$  est arbitraire, par exemple  $\mathbf{a} = \mathbf{o}$ .



Cette courbe dérivée est appelée une *hodographe*.

**Remarque :** distinguons entre espace affine  $\mathbb{E}^d$  et espace vectoriel  $\mathbb{R}^d$ . Les éléments de  $\mathbb{E}^d$  sont des points, tandis que les éléments de  $\mathbb{R}^d$  sont des vecteurs. La soustraction de deux points  $\mathbf{a} - \mathbf{b} = \mathbf{v}$  donne un vecteur. Les points ne peuvent pas être additionnés, par contre les vecteurs le peuvent.

Pour les points les combinaisons barycentriques (=combinaisons affines) (sommes pondérées) existent :  $\mathbf{b} = \sum \alpha_i \mathbf{b}_i$ ,  $\mathbf{b}_i \in \mathbb{E}^d$  et  $\sum \alpha_i = 1$ .

## Dérivées d'ordre supérieur

Notons  $\Delta^r \mathbf{b}_i = \Delta^{r-1} \mathbf{b}_{i+1} - \Delta^{r-1} \mathbf{b}_i$  l'opérateur de différence avant itéré.

Par exemple

$$\begin{aligned}\Delta^0 \mathbf{b}_i &= \mathbf{b}_i \\ \Delta^1 \mathbf{b}_i &= \mathbf{b}_{i+1} - \mathbf{b}_i \\ \Delta^2 \mathbf{b}_i &= \mathbf{b}_{i+2} - 2\mathbf{b}_{i+1} + \mathbf{b}_i \\ \Delta^3 \mathbf{b}_i &= \mathbf{b}_{i+3} - 3\mathbf{b}_{i+2} + 3\mathbf{b}_{i+1} - \mathbf{b}_i\end{aligned}$$

Les coefficients des expressions à droite sont les coefficients du binôme, formant un triangle à celui de Pascal.

Généralement, on a

$$\Delta^r \mathbf{b}_i = \sum_{j=0}^r \binom{r}{j} (-1)^{r-j} \mathbf{b}_{i+j}.$$

La  $r$ -ième dérivée d'une courbe de Bézier :

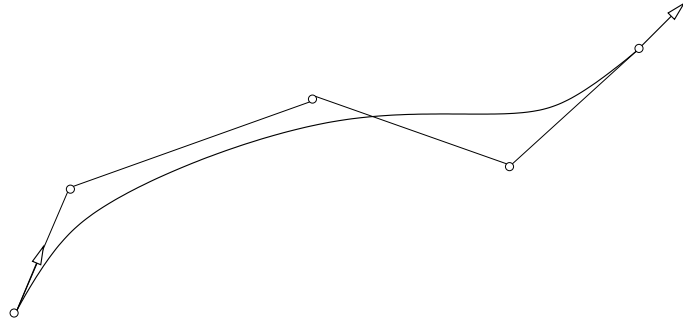
$$\frac{d^r}{dt^r} \mathbf{x}(t) = \frac{n!}{(n-r)!} \sum_{i=0}^{n-r} \Delta^r \mathbf{b}_i B_i^{n-r}(t).$$

Les deux cas particuliers suivants sont importants :

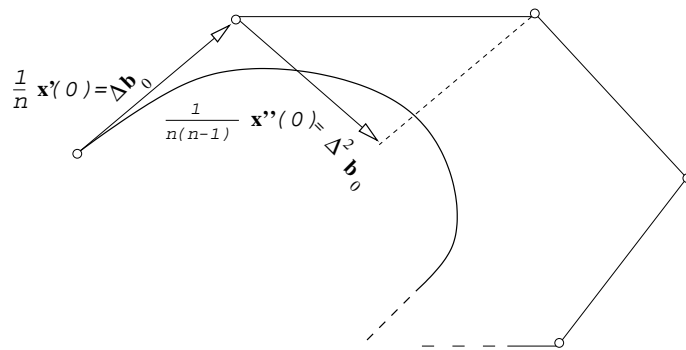
$$\begin{aligned}\frac{d^r}{dt^r} \mathbf{x}(0) &= \frac{n!}{(n-r)!} \Delta^r \mathbf{b}_0, \\ \frac{d^r}{dt^r} \mathbf{x}(1) &= \frac{n!}{(n-r)!} \Delta^r \mathbf{b}_{n-r}.\end{aligned}$$

## Théorème de Bézier

La dérivée  $r$ -ième d'une courbe de Bézier en une de ses extrémités dépend uniquement des  $r+1$  points de contrôle les plus proche de cette extrémité.



$$\begin{aligned}
 x(0) &= b_0 & x(1) &= b_n \\
 x'(0) &= n\Delta b_0 & x'(1) &= n\Delta b_{n-1} \\
 x''(0) &= n(n-1)\Delta^2 b_0 & x''(1) &= n(n-1)\Delta^2 b_{n-2}
 \end{aligned}$$

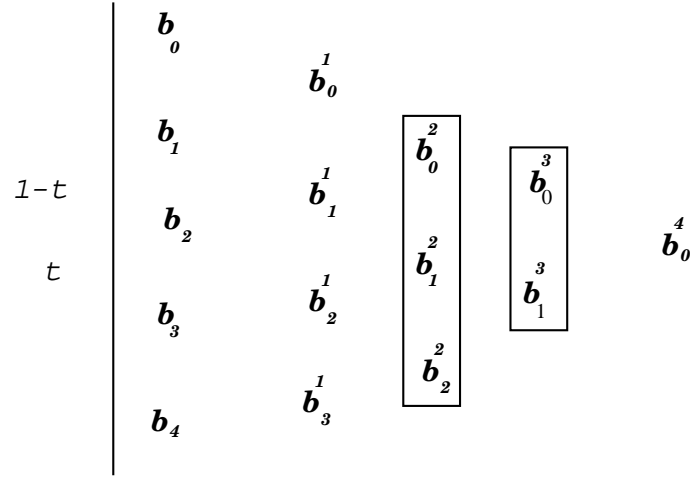


## Dérivées et algorithme de de Casteljau

Les dérivées d'une courbe de Bézier peuvent être exprimées en fonction des points intermédiaires générés par l'algorithme de de Casteljau :

$$\frac{d^r}{dt^r} x(t) = \frac{n!}{(n-r)!} \Delta^r b_0^{n-r}(t).$$

**Exemple :**  $n = 4$



$$\mathbf{x}'(t) = 4 \cdot \Delta \mathbf{b}_0^3 = 4(\mathbf{b}_1^3 - \mathbf{b}_0^3)$$

$$\mathbf{x}''(t) = 4 \cdot 3 \Delta^2 \mathbf{b}_0^2 = 12(\mathbf{b}_2^2 - 2\mathbf{b}_1^2 + \mathbf{b}_0^2)$$

— dessin —

Ceci est possible, car la sommation et l'opérateur différence commutent :

$$\sum_{j=0}^{n-r} \Delta \mathbf{b}_j = \sum_{j=0}^{n-r} \mathbf{b}_{j+1} - \sum_{j=0}^{n-r} \mathbf{b}_j = \Delta \sum_{j=0}^{n-r} \mathbf{b}_j$$

grâce à ceci, on a

$$\begin{aligned} \frac{d^r}{dt^r} \mathbf{x}(t) &= \frac{n!}{(n-r)!} \sum_{i=0}^{n-r} \Delta^r \mathbf{b}_i B_i^{n-r}(t) \\ &= \frac{n!}{(n-r)!} \Delta^r \sum_{i=0}^{n-r} \mathbf{b}_i B_i^{n-r}(t) \\ &= \frac{n!}{(n-r)!} \Delta^r \mathbf{b}_0^{n-r}. \end{aligned}$$

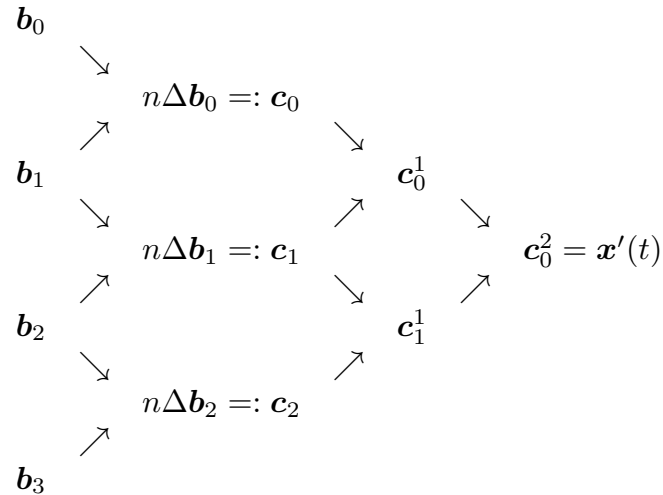
Ces équations (la première et la dernière) suggèrent deux différentes méthodes de calcul de la  $r$ -ième dérivée d'une courbe de Bézier :



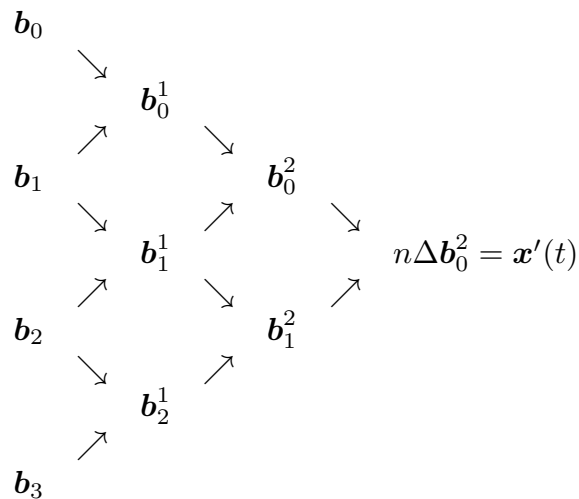
### Méthode 1 :

Calculer toutes les différences  $r^{\text{ème}}$  vers l'avant des points de contrôle, puis les interpréter comme un nouveau polygone de contrôle de degré  $n - r$  et l'évaluer en  $t$ .

**Exemple :**  $n = 3$  et  $r = 1$



### Méthode 2 :



L'algorithme de de Casteljau offre une solution rapide du problème d'obtenir un point sur la courbe et de connaître la dérivée première et/ou seconde en même temps.

Pour obtenir la dérivée  $r^{\text{ième}}$ , il faut effectuer  $r$  opérations de différence et  $(n - r)$  opérations d'évaluation.