

Introduction:

Les objets fabriqués par l'industrie comportent des lignes courbes. Les logiciels de CAO proposent un catalogue de formes simples (segments de droites, arcs de cercles, de coniques...) mais elles ne suffisent pas. Le concepteur a besoin d'une famille plus riche de courbes.

Les machines à commande numériques ne savaient usiner de façon précise que formes simple. Une seconde catégorie d'objets, au contraire, aurait une forme a priori peu précise, déterminée expérimentalement. Les hélices d'avions, les coques de bateaux et les carrosseries de voitures étaient tracées à main levée, sans que l'on puisse décrire leurs formes par une formule mathématique.

le problème auquel on est souvent confronté est le suivant : comment représenter, d'un point de vue informatique, des courbes ou des surfaces de "forme libre", c'est à dire des courbes ou des surfaces continues, d'allure globalement régulière, dont la forme n'est régie, à priori, par aucune équation particulière ?

Mode de représentation :

Nous récapitulons ici les principaux types d'équations associées à la notion de courbe ou de surface. Les lettres x, y et z, de même que les illustrations, suggèrent l'emploi d'un repère cartésien, mais rien n'empêche, bien entendu, de recourir à d'autres systèmes de coordonnées.

Equations implicites :

- Courbes 2D : $f:\mathbb{R}^2 \rightarrow \mathbb{R} \quad f(x,y)=0$

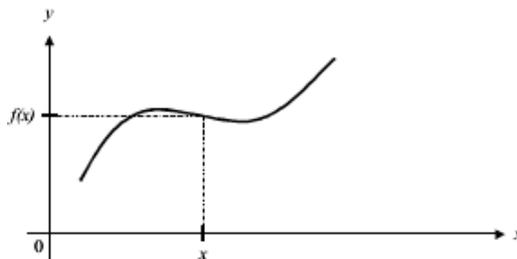
- Courbes 3D : $f:\mathbb{R}^3 \rightarrow \mathbb{R}^2 \quad f(x,y)=\begin{bmatrix} 0 \\ 0 \end{bmatrix}$

- Surface : $f:\mathbb{R}^3 \rightarrow \mathbb{R} \quad f(x,y,z)=0$

Fonctions réelles d'une ou deux variables :

- Courbes 2D :

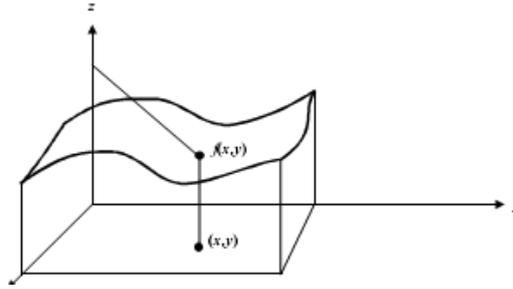
$$f:\mathbb{R} \rightarrow \mathbb{R} \quad y=f(x)$$



Chapitre II : Modélisation des courbes

- Surface :

$$f: \mathbb{R}^2 \rightarrow \mathbb{R} \quad z = f(x, y)$$



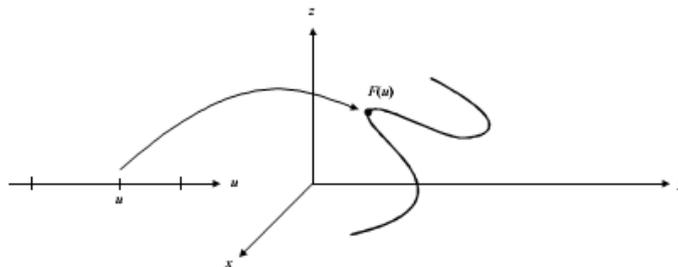
Courbes et surfaces paramétrées :

- Courbes 2D :

$$F: I \subset \mathbb{R} \rightarrow \mathbb{R}^2 \quad (x, y) = f(u)$$

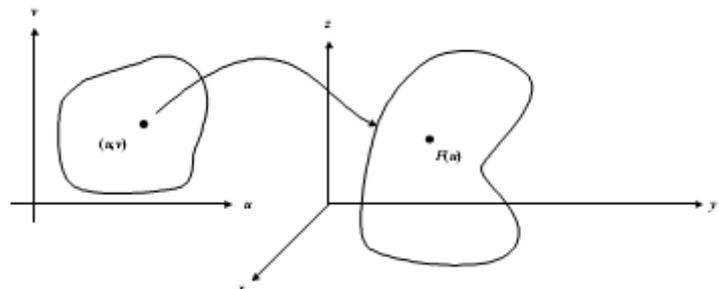
- Courbes 3D :

$$F: I \subset \mathbb{R} \rightarrow \mathbb{R}^3 \quad (x, y, z) = f(t)$$



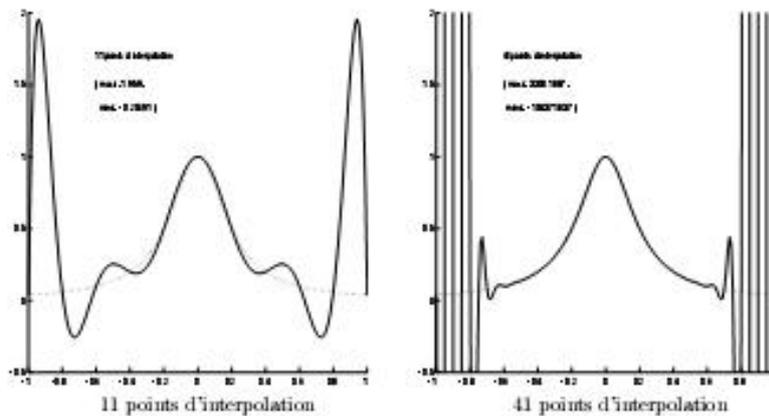
- Surface :

$$F: I \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3 \quad (x, y, z) = f(u, v)$$



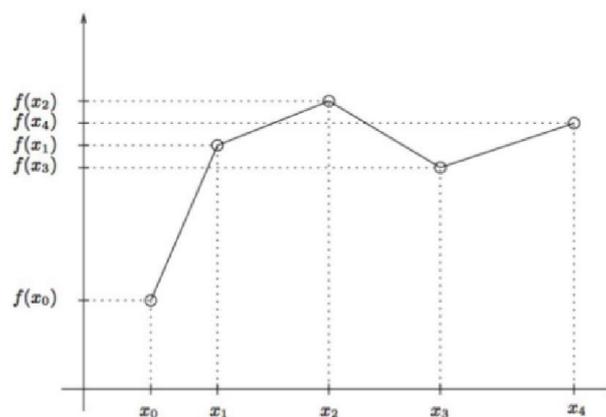
Approche « polynômes par morceaux » :

L'utilisation de polynômes de degré élevé est délicate et mène parfois à des oscillations de grande amplitude. Les polynômes de degré élevé sont alors peu adéquats. On peut mesurer la régularité d'une fonction par le biais de ses dérivées.



Le but de cette approche est de conserver la forme polynomiale, mais de faire en sorte de ne pas augmenter le degré du polynôme. L'idée pour cela est de fabriquer une fonction qui soit une succession de polynômes de degré peu élevé (degré 3 dans le cas général) tels que les raccordements entre polynômes soient le plus réguliers possible.

Illustrons d'abord cette idée par un exemple simple appelé "interpolation linéaire par morceaux". On utilise aussi l'appellation splines linéaires.

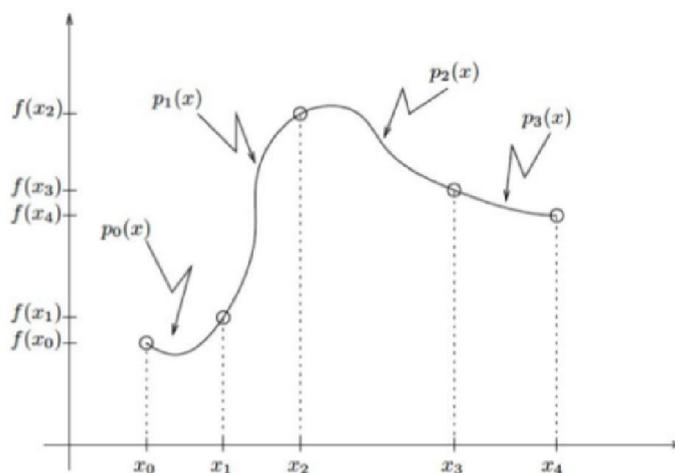


On imagine assez mal comment une telle courbe pourrait permettre de faire la conception d'une carrosserie de voiture ou d'une aile d'avion. Il faut donc être plus prudent à la jonction des différents segments de courbe. La spline linéaire est continue mais n'est pas dérivable et nous allons maintenant montrer que l'on peut faire beaucoup mieux.

Les splines cubiques représentent un compromis très intéressant entre la régularité de la courbe obtenue et le degré des polynômes utilisés.

Interpolation par spline cubique :

On considère un ensemble de points (x_0, y_0) , (x_1, y_1) , (x_{i-1}, y_{i-1}) , (x_i, y_i) , (x_{i+1}, y_{i+1}) , ... (x_{i-1}, y_{i-1}) . Pour réaliser une interpolation entre ces points avec une spline cubique, un polynôme du troisième degré est utilisé entre chaque deux points.



L'équation à droite d'un point (x_i, y_i) est f_i avec une valeur y de $f_i(x_i)$ au point x_i . De même l'équation à gauche d'un point (x_i, y_i) est f_{i+1} avec une valeur y de $f_{i+1}(x_i)$ au point x_i .

La fonction spline cubique f_i est construite sur la base des critères suivants :

- Les courbes sont des polynômes du troisième degré :

$$f_i(x) = a_i + b_i x + c_i x^2 + d_i x^3 \quad - (1)$$

- les courbes passent à travers tous les points :

$$f_i(x_i) = f_{i+1}(x_i) = y_i \quad - (2)$$

- La tangente ou la première dérivée est la même pour les deux fonctions des deux côtés d'un point.

$$f'_i(x_i) = f'_{i+1}(x_i) \quad - (3)$$

- La dérivée seconde est la même dans les deux côtés d'un point :

$$f''_i(x_i) = f''_{i+1}(x_i) \quad - (4)$$

Chapitre II : Modélisation des courbes

Le résultat est une matrice pour n-1 équation est n+1 inconnus. Les deux équations qui manquent sont déterminés sur la base des conditions aux limites pour le point de départ $f_1(x_0)$ et le point d'arrivée $f_n(x_n)$. Historiquement une de ces conditions sont utilisés.

Spline cubique naturelle : les dérivés seconds pour la spline sont considérées nulles aux extrémités :

$$f_1''(x_0) = f_n''(x_n) = 0 \quad - (5a)$$

Spline Parabolique : les dérivés seconds aux extrémités sont égales à celles des points adjacents. Le résultat est que la courbe devient parabolique à ces extrémités.

$$\begin{aligned} f_1''(x_0) &= f_1''(x_1) \\ f_n''(x_n) &= f_n''(x_{n-1}) \end{aligned} \quad - (5b)$$

Spline Cubique dite « not-a-knot » : la courbe se dégrade vers une seule courbe cubique sur les derniers segments des deux extrémités. Pour cela ont impose les dérivés seconds égales à :

$$\begin{aligned} f_1'''(x_0) &= 2f_1'''(x_1) - f_2'''(x_2) \\ f_n'''(x_n) &= 2f_n'''(x_{n-1}) - f_{n-1}'''(x_{n-2}) \end{aligned} \quad - (5c)$$

D'habitude les équations 2 à 5 sont combinées pour obtenir un système d'équation tridiagonale n+1 par n+1 dont la résolution nous donne les paramètres pour chaque segment de la spline cubique.

$$\left(\frac{1}{3}\right) f_0'' + 2f_1'' + \left(\frac{2}{3}\right) f_2'' = 6 \left(-\frac{1}{6}\right)$$

et la deuxième équation ($i = 1$) s'écrit :

$$\left(\frac{2}{3}\right) f_1'' + 2f_2'' + \left(\frac{1}{3}\right) f_3'' = 6 \left(-\frac{1}{6}\right)$$

Pour obtenir la spline naturelle ($f_0'' = f_3'' = 0$), on résout le système :

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{3} & 2 & \frac{2}{3} & 0 \\ 0 & \frac{2}{3} & 2 & \frac{1}{3} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_0'' \\ f_1'' \\ f_2'' \\ f_3'' \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ -1 \\ 0 \end{bmatrix}$$

dont la solution est :

$$f_0'' = 0, f_1'' = -3/8, f_2'' = -3/8 \text{ et } f_3'' = 0.$$

Pour obtenir l'équation de la spline dans le premier intervalle, on doit utiliser les relations 6 et 11. On obtient :

$$f_0 = 1$$

$$f_0' = f[x_0, x_1] - \frac{h_0 f_0''}{3} - \frac{h_0 f_1''}{6} = 3 - \frac{(1)(0)}{3} - \frac{(1)(-3/8)}{6} = \frac{49}{16}$$

$$f_0''' = \frac{f_1'' - f_0''}{h_0} = -\frac{3}{8}$$

et on a :

$$p_0(x) = 1 + \frac{49}{16}(x-1) + \frac{0}{2}(x-1)^2 - \frac{3}{48}(x-1)^3$$

Ce polynôme n'est défini que dans l'intervalle $[1, 2]$. On peut par exemple l'évaluer en $x = 1,5$ pour obtenir 2,523 4375.

Avantages et inconvénients :

Les courbes spline sont très harmonieuses. Il n'y a effectivement pas de cassure du rayon de courbure. C'est pourquoi l'interpolation par splines cubiques peut être utilisée dans le cadre de la CAO.

Les splines d'interpolation présentent aussi des inconvénients:

- il n'est pas possible de contrôler localement une spline car la modification des coordonnées d'un point influe sur toute la courbe

- La spline peut-elle aussi devenir oscillante si les dérivées de la fonction à interpoler deviennent trop grandes ($\gg 1$).
- la spline dépend du choix du système de coordonnées, donc elle ne possède pas une propriété d'invariance géométrique.

Les Courbes de Bézier :

A cause des inconvénients cités plus haut, l'interpolation polynomiale par spline n'est pas toujours très adaptée pour certaines utilisations en CAO plus précisément dans l'industrie automobile. Dans les années 1960, les machines à commandes numériques sont apparues, il fallait donc décrire les formes (comme les courbes de carrosserie) avec des équations mathématiques.

La première solution était d'interpoler linéairement un grand nombre de points. Cette méthode a de nombreux inconvénients :

- Pour la machine, il y a beaucoup de paramètres.
- Il est impossible d'agrandir (mais aussi de translater, de déformer, . . .) une partie d'une pièce sans rajouter de points supplémentaires.
- Placer des points n'est pas intuitif pour les designers.
- Il est très fastidieux de modifier la courbe

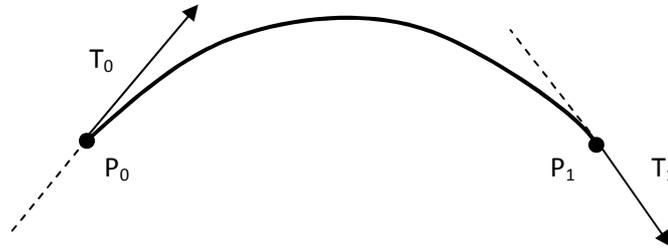
Un autre procédé était donc nécessaire pour exprimer une courbe avec peu de paramètres et que ceux ci soient naturels.

L'idée révolutionnaire des courbes de Bézier est l'utilisation de points de contrôle et non de points d'interpolation. Cela veut dire que la courbe ne passe pas par les points donnés mais les approche. Les courbes de Bézier ne sont donc pas des interpolations mais des approximations. Il y a plusieurs avantages à cela :

- La courbe est stable, il est facile de déformer la courbe sans résultats inattendus.
- Il est facile de modifier la courbe, il ne faut que modifier les points de contrôle qui sont peu nombreux.
- Le placement des points de contrôle est relativement évident.
- Il est plus facile d'avoir une courbe naturelle, fluide, non brusquée avec des points de contrôle qu'avec des points d'interpolation.
- Et tout cela en gardant les avantages du modèle numérique (agrandissement, déformation, etc.).

Courbes de Hermite cubiques :

C'est des courbes polynomiales de degré 3 définies par la position et la tangente aux points de départ et d'arrivée.



$$C(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + a_3 \cdot x^3$$

Sous forme paramétrique :

$$C(t) = \begin{cases} x(t) = a_0 + a_1 \cdot t + a_2 \cdot t^2 + a_3 \cdot t^3 \\ y(t) = b_0 + b_1 \cdot t + b_2 \cdot t^2 + b_3 \cdot t^3 \\ z(t) = c_0 + c_1 \cdot t + c_2 \cdot t^2 + c_3 \cdot t^3 \end{cases} \quad t \in [0,1]$$

$$C'(t) = \begin{cases} x(t) = a_1 + 2a_2t + 3a_3t^2 \\ y(t) = b_1 + 2b_2t + 3b_3t^2 \\ z(t) = c_1 + 2c_2t + 3c_3t^2 \end{cases}$$

Les contraintes à respecter :

1. La courbe passe par le point P_1 : $C(0)=P_1$
2. La courbe passe par le point P_2 : $C(1)=P_2$
3. La tangente au point P_1 égale à T_1 : $C'(0)=T_1$
4. La tangente au point P_2 égale à T_2 : $C'(1)=T_2$

En appliquant ces contraintes pour la coordonnée x on aboutit au système d'équations suivant :

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} p_{0x} \\ p_{1x} \\ T_{0x} \\ T_{1x} \end{bmatrix}$$

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} p_{0x} \\ p_{1x} \\ T_{0x} \\ T_{1x} \end{bmatrix}$$

Ce qui donne :

$$x(t) = (1 - 3t^2 + 2t^3)p_{0x} + (3t^2 - 2t^3)p_{1x} + (t - 2t^2 + t^3)T_{0x} + (-t^2 + t^3)T_{1x}$$

En effectuant le même raisonnement pour les coordonnées y et z on obtient :

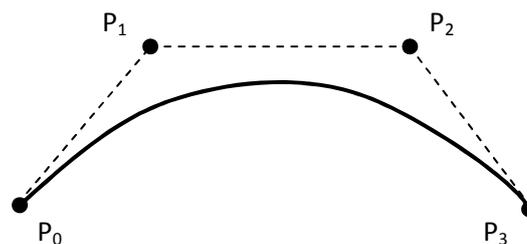
$$C(t) = (1 - 3t^2 + 2t^3)p_0 + (3t^2 - 2t^3)p_1 + (t - 2t^2 + t^3)T_0 + (-t^2 + t^3)T_1$$

Courbe de Bézier cubiques :

C'est des courbes polynomiales de degré 3 définies par les positions de quatre points.

Dans ce cas, les tangentes au premier et au dernier point seront déduites par les positions respectives des deux premiers et des deux derniers points.

$$T_0 = 3(p_1 - p_0), T_1 = 3(p_3 - p_2)$$



Le résultat est la formulation suivante :

$$C(t) = (1 - t)^3 p_0 + 3t(1 - t)^2 p_1 + 3t^2(1 - t)p_2 + t^3 p_3$$

Sous forme matricielle :

$$C(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix}$$

Qui peut être aussi mise sous la forme :

$$C(t) = B_0(t)p_0 + B_1(t)p_1 + B_2(t)p_2 + B_3(t)p_3$$

Avec :

$$B_0(t) = (1 - t^3)$$

$$B_1(t) = 3t(1 - t)^2$$

$$B_2(t) = 3t^2(1 - t)$$

$$B_3(t) = t^3$$

Et courbe de Bézier cubique peut être exprimée aussi sous forme compacte comme suit :

$$C(t) = \sum_{i=0}^3 P_i B_{i,3}(t)$$

Les fonctions $B_{i,3}(t)$ sont les fonctions de Bernstein cubique.

La forme générale de ces fonctions à un degré n est donnée par l'expression suivante :

$$B_{i,n}(t) = \left(\frac{n!}{i!(n-i)!} \right) t^i (1-t)^{n-i}$$

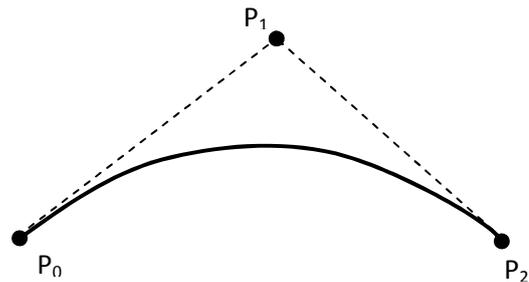
L'expression générale d'une courbe de Bézier de degré n devient :

$$C(t) = \sum_{i=0}^n P_i B_{i,n}(t)$$

Pour une courbe de Bézier quadratique (degré 2) :

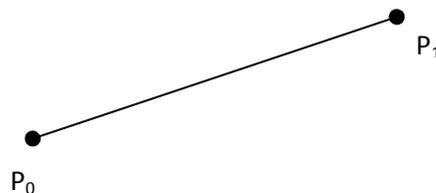
$$C(t) = \sum_{i=0}^2 P_i B_{i,2}(t)$$

$$C(t) = (1-t)^2 p_0 + 2t(1-t)p_1 + t^2 p_2$$



Une courbe de Bézier du premier degré :

$$C(t) = (1-t)p_0 + tp_1$$



Détermination d'un point sur une courbe de Bézier:

Pour déterminer les coordonnées (x, y, z) pour une valeur donnée du paramètre t sur une courbe de Bézier de degré n il est toujours possible d'utiliser la formulation générale mais cette dernière est rarement utilisée dans la pratique car elle peut engendrer des erreurs d'approximation lors de sa programmation.

Dans la pratique on utilise, le plus souvent, une méthode basée sur l'**algorithme de De Casteljau** du nom du mathématicien Paul De Casteljau qui travailla à Citroën lui aussi sur la définition des formes libres depuis 1959 mais ces travaux ont été gardés secrets jusqu'à 1975. Cet algorithme est basé sur l'idée suivante : si on prend une courbe de Bézier, de degré 3 par exemple, cette dernière est une combinaison de deux courbes quadratiques

$$C(t) = (1-t^3)p_0 + 3t(1-t)^2 p_1 + 3t^2(1-t)p_2 + t^3 p_3$$

$$C(t) = (1-t) \underbrace{((1-t)^2 p_0 + 2t(1-t)p_1 + t^2 p_2)}_{\text{quadratique}} + t \underbrace{((1-t)^2 p_1 + 2t(1-t)p_2 + t^2 p_3)}_{\text{quadratique}}$$

Cette idée peut être généralisée pour une courbe de degré n qui peut donner dans ce cas deux courbes de degré n-1 comme suit :

$$C^n(p_0, \dots, p_n) = (1-t)C^{n-1}(p_0, \dots, p_{n-1}) + tC^{n-1}(p_1, \dots, p_n)$$

Chapitre II : Modélisation des courbes

Cette expression peut être réécrite en fonction des points de contrôle pour obtenir la formulation suivante :

$$P_i^k(t_0) = (1 - t_0)P_i^{k-1}(t_0) + t_0P_{i+1}^{k-1}(t_0), \begin{cases} k = 1, \dots, n \\ i = 0, \dots, n - k \end{cases}$$

Nous pouvons vérifier avec cette formulation que :

$$C(t_0) = P_0^n(t_0)$$

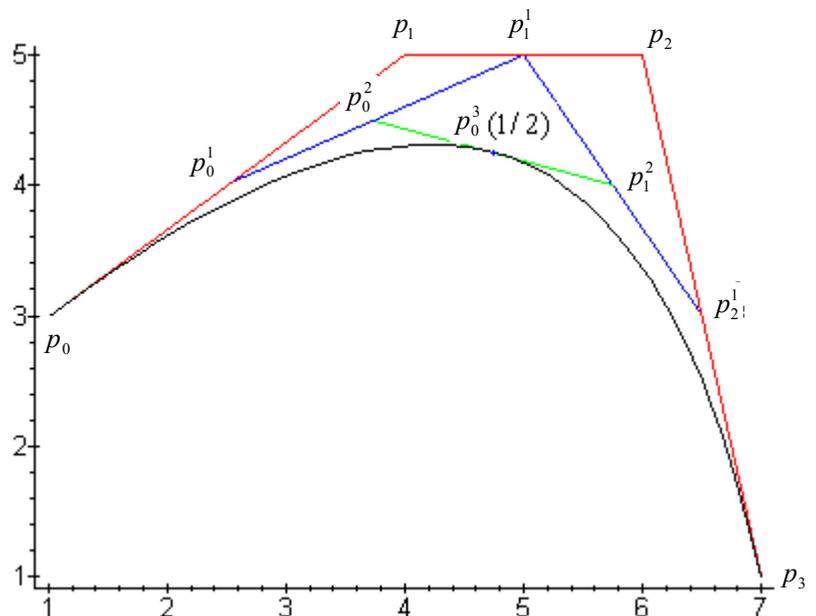
Prenons un exemple : dans le cas où n=3

On prend un plan muni d'un repère (O, \vec{i}, \vec{j}) Soient :

- $P_0(1,3)$
- $P_1(4,5)$
- $P_2(6,5)$
- $P_3(7,1)$
- $t \in [0,1]$

Les coefficients intermédiaires peuvent être écrits sous forme d'un tableau triangulaire de points, appelé schéma de De Casteljau :

$$\begin{array}{cccc} P_0 & & & \\ P_1 & P_0^1 & & \\ P_2 & P_1^1 & P_0^2 & \\ P_3 & P_2^1 & P_1^2 & P_0^3 \end{array}$$



Inconvénients des courbes de Bézier :

Les courbes de Bézier ont beaucoup d'avantages pour une utilisation pratique. Néanmoins, ils ont quelques inconvénients :

- La courbe ne passant pas par les points de contrôle, il peut être difficile de contrôler la courbe bien que, comme nous l'avons vu, l'utilisation de points de contrôle facilite souvent la conception. Ainsi, les courbes de Bézier ne seraient pas très efficaces pour tracer une courbe de tendance sur des mesures. Pour ce type de problèmes, nous nous tournons plus vers l'interpolation cubique et par splines cubiques.
- Un autre inconvénient est le manque de contrôle local, le décalage d'un point fait bouger toute la courbe. Dans l'industrie automobile, par exemple, il est gênant que toute la pièce change de forme lorsque nous voulons seulement faire varier une partie de la pièce. Au niveau des calculs, il faudra calculer à nouveau toute la pièce.
- Le troisième inconvénient est le degré des courbes. En effet, pour une forme complexe, nous devons utiliser beaucoup de points de contrôle donc le degré de la courbe est élevé.

Les B-Splines :

Il s'agit de construire une courbe qui présente tous les avantages des courbes de Bézier mais sans ses inconvénients. Ainsi, la courbe devra approximer les points de contrôle, être simple à manipuler, présenter les mêmes propriétés que les courbes de Bézier, etc. Le degré de la courbe ne devra pas être proportionnel au nombre de points de contrôle mais. La modification d'un point ne doit pas affecter toute la courbe.

Les B-Splines ont été développée chez Boeing dans les années 70 et 80. L'idée principale des B-Splines est de remplacer les polynômes de Bernstein par des fonctions. Ensuite, nous sommerons ces fonctions avec les points de contrôles pour obtenir la courbe. Une B-Spline ne dépend pas uniquement des points de contrôle mais aussi d'un vecteur nœud.

Courbes B-Splines :

Une courbe B-Spline est définie par :

$$C(t) = \sum_{i=0}^n P_i N_{i,k}(t)$$

Chapitre II : Modélisation des courbes

Avec :

- P_i ($i=1, \dots, n-1$) sont les points de contrôle.
- $N_{i,k}$ les fonctions de base de B-Spline

Pour $k=1$:

$$N_{i,1}(t) = \begin{cases} 1 & \text{if } t \in [t_i, t_{i+1}) \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Pour $k>1$:

$$N_{i,k}(t) = \left(\frac{t - t_i}{t_{i+k-1} - t_i} \right) N_{i,k-1}(t) + \left(\frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} \right) N_{i+1,k-1}(t), \quad t \in [k-1, n+1) \quad (2)$$

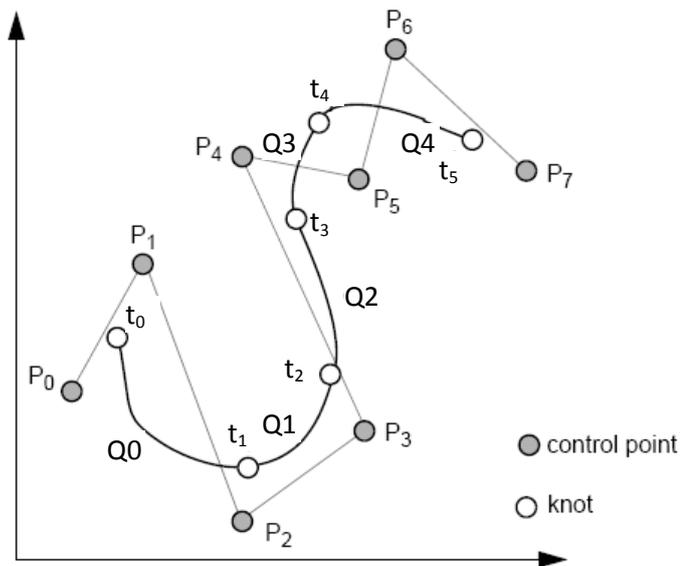
- k est l'ordre des segments de la courbe B-Spline. Un ordre k veut dire que la courbe est composée de k segments de polynômes de degré $(k-1)$. Et donc le degré de la B-Spline ne dépend pas du nombre de points de contrôle.
- t_i ($i=0, \dots, n+k$) est l'ensemble de nœuds qui est une séquence croissante de nombre réels. Le vecteur de nœuds peut être :

Uniforme : où les éléments du vecteur sont espacés avec un pas constant.

Exemples : $\{0, 1, 2, 3, 4, 5, 6, 7\}$, $\{-0,2, -0,1, 0, 0,2, 0,1\}$, $\{0, 0,25, 0,5, 0,75, 1\}$

Non Uniforme : où l'intervalle d'espacement entre les éléments du vecteur est différent. Exemples : $\{0, 0, 0, 1, 1, 2, 2, 2\}$, $\{0, 0,25, 0,75, 0,8, 1\}$, $\{0, 1, 1, 1, 2, 3, 4, 5\}$,

Si $t_i = t_{i+1}$ alors le coefficient $\frac{t - t_i}{t_{i+k} - t_i}$ nous aurons $\begin{pmatrix} \dots \\ 0 \end{pmatrix}$, nous posons ce quotient égale à 0.



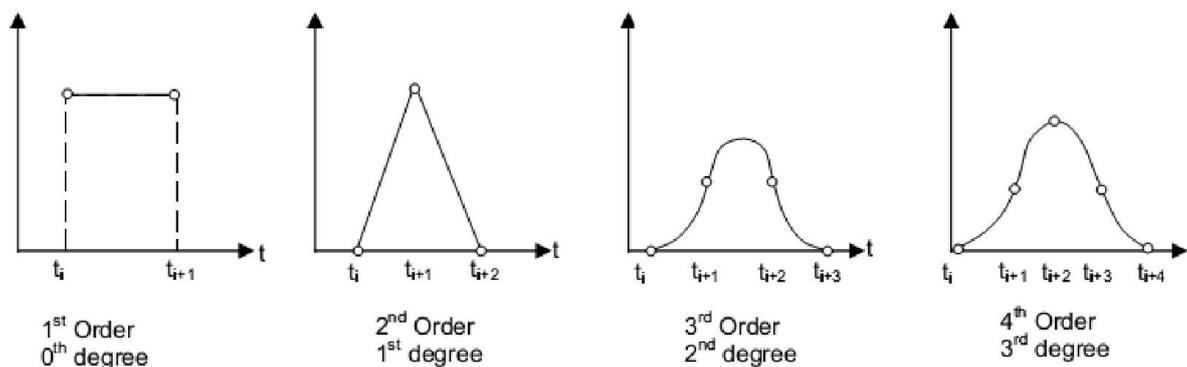
Remarques :

- Courbe définie par morceaux de degré $k-1$
- Chaque morceau défini sur $[t_i, t_{i+1}]$
- Chaque $N_{i,k}$ a un support $[t_i, t_{i+k}]$
- Donc :
 - $N_{0,k}(t)$ a un support $[t_0, t_k]$
 - $N_{n,k}(t)$ a un support $[t_n, t_{n+k}]$
- Par conséquent : il doit y avoir $(n + k + 1)$ nœuds, le vecteur nodal est alors $[t_0, t_1, \dots, t_{n+k}]$
- La modification : Soit un point de contrôle P_i et $N_{i,k}(t)$ la fonction de base associée. Alors, si on déplace P_i on ne modifie que la partie de la courbe pour laquelle $t \in [t_i, t_{i+k}]$. Donc la modification est locale alors qu'elle est globale pour la courbe de Bézier.
- Chaque morceau de courbe est défini sur un intervalle $[t_i, t_{i+1}]$.
- Chaque fonction de base $N_{i,k}(t)$ couvre k intervalles.
- Il y a au plus k $N_{i,k}(t)$ non nulles sur un intervalle.
- Donc chaque morceau de courbe est défini par k points de contrôle :
 - le 1^{er} morceau : $(P_0, P_1, \dots, P_{k-1})$
 - le dernier morceau : (P_{n-k+1}, \dots, P_n)

- $N_{n,k}(t)$ a comme support $[t_i, t_{i+1}]$.

Graph des fonctions $N_{i,k}$:

Pour l'ordre 1 les fonctions de base auront des valeurs constantes sur leurs supports. La fonction d'ordre 2 est composée de deux segments de droite qui peuvent être exprimé par des polynômes de degré 1. Pour l'ordre 3 qui est le plus souvent utilisé les fonctions N sont composées de trois segments de degré 2 deux hyperboles et une parabole au milieu.



Support ou intervalle des fonctions de base :

Nous avons vu sur les graphes qu'une fonction $N_{i,k}(t)$ s'annule en dehors de l'intervalle $[t_i, t_{i+k}[$. Nous pouvons démontrer récursivement que par exemple $N_{i,3}(t)$ s'obtient à partir de $N_{i,1}(t)$, $N_{i+1,1}(t)$, $N_{i+2,1}(t)$, $N_{i+3,1}(t)$. et nous avons défini que : $N_{i,1}(t) = 1_{[t_i, t_{i+1}[}$. Donc $N_{i,3}(t)$ est bien nulle en dehors de l'intervalle $[t_i, t_{i+3}[$. De la même façon nous pouvons démontrer $N_{i,k}(t)$ s'annule en dehors de l'intervalle $[t_i, t_{i+k}[$.

Les B-splines cubiques uniformes :

Les paramètres t_i sont uniformément répartis sur l'ensemble des points de contrôle, à savoir :

$$t_i = t_{i-1} + 1.$$

Donc t varie de 1 pour un segment de courbe et on effectue le changement de variable $t=t-t_i$. La position d'un point P est alors définie par :

$$P(t) = \frac{1}{6}[(1-t)^3 P_{i-3} + (3t^3 - 6t^2 + 4)P_{i-2} + (-3t^3 + 3t^2 + 3t + 1)P_{i-1} + t^3 P_i],$$

Pour : $0 \leq t < 1$.

Soit sous forme matricielle :

$$P(t) = (t^3, t^2, t, 1) \cdot \frac{1}{6} \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} P_{i-3} \\ P_{i-2} \\ P_{i-1} \\ P_i \end{pmatrix}$$

$$P(t) = T^t \cdot M_{bspline} \cdot P.$$

Avantages et inconvénients des B-spline :

Les B-Splines résolvent les problèmes que nous avons constaté pour les courbes de Bézier, il est possible de contrôler les B-Splines localement et l'ajout de points n'augmente pas le degré de la courbe. De plus, il est possible d'interpoler les points de contrôle même si ce n'est pas très concluant.

Le principal inconvénient des B-Splines est leur complexité. En effet, il n'est pas facile de calculer les fonctions de base. Les points de contrôle ne sont plus les seuls paramètres des courbes, il y a aussi le vecteur nœud. Il est difficile de gérer les points et les nœuds en même temps. C'est pourquoi nous ne faisons que varier les points de contrôle.

Courbes polynomiales rationnelles :

Les segments de courbes rationnelles sont des rapports de polynômes :

$$x(t) = X(t)/W(t), \quad y(t) = Y(t)/W(t), \quad z(t) = Z(t)/W(t).$$

Où $X(t), Y(t), Z(t), W(t)$ sont des courbes polynomiales dont les points de contrôle sont définis en coordonnées homogènes. L'ensemble des définitions précédemment introduites s'appliquent ici en rajoutant une troisième (2D) ou quatrième (3D) coordonnées aux points traités. On peut donc calculer des Bézier rationnelles ou des B-splines rationnelles non-uniformes (NURBS). L'intérêt des courbes rationnelles est double :

1. les courbes rationnelles sont invariantes par rotation, changement d'échelle, translation et projection perspective. Ce qui veut dire qu'une transformation peut-être appliquée aux points de contrôle uniquement.
2. Un deuxième intérêt est que les splines rationnelles (quadratiques) permettent de tracer précisément les coniques sans avoir besoin de nombreux points de contrôle.

Les Courbes NURBS :

Les seules différences entre les NURBS et les B-Splines sont le type de vecteur de nœuds choisi et le vecteur de poids w_i .

La formulation d'une courbe NURBS est la suivante :

$$C(t) = \sum_{i=0}^n \frac{N_{i,k} w_i}{\sum_{j=0}^n N_{j,k} w_j} P_i = \frac{\sum_{i=1}^n N_{i,k} w_i P_i}{\sum_{i=1}^n N_{i,k} w_i}$$

Qui peut être mise sous la forme :

$$C(t) = \sum_{i=0}^n R_{i,k}(t) P_i$$

Avec

$$R_{i,k}(t) = \frac{\sum_{i=1}^n N_{i,k} w_i P_i}{\sum_{i=1}^n N_{i,k} w_i}$$

Le vecteur de nœuds pour les NURBS est de type non-uniforme, c'est-à-dire que l'espacement entre deux valeurs de nœuds consécutifs n'est pas constant.

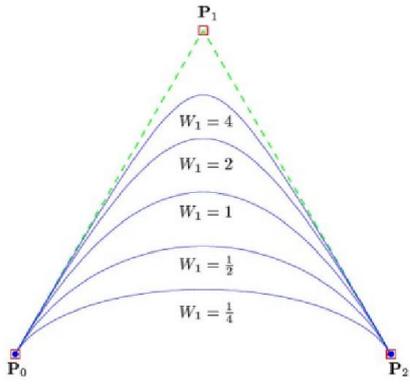
À titre, d'exemple :

$$\begin{aligned} \mathbf{t} &= [t_1 \ t_2 \ t_3 \ \dots \ t_{n+d-1} \ t_{n+d} \ t_{n+d+1}] \\ &= [1 \ 1.5 \ 5 \ 8 \ 9 \ 11 \ \dots \ n+d-1 \ n+d \ n+d+1] \end{aligned}$$

Toutefois, la définition des fonctions splines restent la même, à la seule différence près que nous ne pouvons pas exprimer les fonctions splines résultantes sous forme polynomiale et ainsi, sous forme matricielle.

En ce qui concerne le vecteur de poids w_i , chaque valeur du vecteur est associée à un point de contrôle P_i . La valeur du poids affecte l'effet qu'à le point de contrôle sur la forme du patch (segment) auquel il est associé.

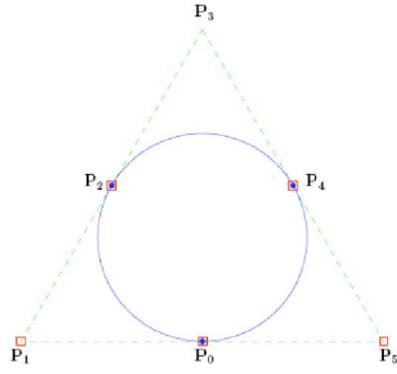
Cet effet peut être comparé à un aimant : si $w_i > 1$, le patch auquel appartient le point de contrôle P_i associé se rapproche du point de contrôle. Si $w_i < 0$, le patch auquel appartient le point de contrôle P_i associé s'éloigne du point de contrôle. Finalement, si $w_i = 1$, aucun effet n'est appliqué sur le patch auquel appartient le point de contrôle P_i associé.



$$\mathbf{T} = (0, 0, 0, 1, 1, 1)$$

$$\mathbf{W} = (1, W_1, 1)$$

$$\mathbf{P} = \{(0, 0); (1, \sqrt{3}); (2, 0)\}$$



$$\mathbf{T} = (0, 0, 0, 1, 1, 2, 2, 3, 3, 3)$$

$$\mathbf{W} = (1, \frac{1}{2}, 1, \frac{1}{2}, 1, \frac{1}{2}, 1)$$

$$\mathbf{P} = \{(1, 0); (0, 0); (\frac{1}{2}, \frac{\sqrt{3}}{2}); (1, \sqrt{3}); (\frac{3}{2}, \frac{\sqrt{3}}{2}); (2, 0); (1, 0)\}$$