

Cours Approximation Géométrique

Géraldine Morin, Sandrine Mouysset

24 mars 2020

CTD2.2 + CTD3 : Courbes de Bézier

1 Approximation, courbes de Bézier

Motivations

Dans cette partie, nous allons voir encore des polynômes mais cette fois ci approximants. Ils permettent de définir des courbes de façon intuitive : les courbes de Bézier.

Compétences

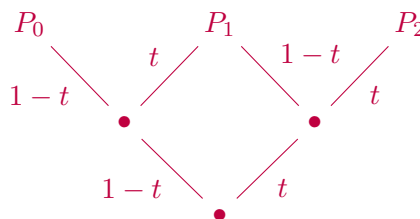
Après avoir vu les deux prochains CTDs, vous devez répondre "oui" à ces questions :

- je sais construire une courbe de Bézier à partir de ses points de contrôle ;
- je connais la définition analytique d'une courbe de Bézier et sais démontrer 5 propriétés d'une courbe de Bézier ;
- je sais produire des approximations linéaires par morceaux des courbes de Bézier, et évaluer la qualité de l'approximation proposée ;
- je sais appliquer l'algorithme de De Casteljau pour évaluer un point d'une courbe de Bézier, ou subdiviser une courbe de Bézier ;
- je sais définir et caractériser la floraison d'un polynôme ;
- je sais détecter qu'une courbe de Bézier a une mauvaise paramétrisation.

Activité préalable

La forme Avant chaque séance, je vous fais passer une petite question avant le CTD. Il faut prendre environ 15mn avant le CTD pour réfléchir à la question posée tout seul ou à plusieurs personnes, mais chacun vient avec une trace écrite de ses réflexions.

La question : Simplifions Neville A partir de 3 points P_0, P_1, P_2 , simplifiez l'algorithme de Neville en utilisant 'partout' les coefficients $1 - t$ et t , comme ci dessous.



Pouvez vous tracer la courbe ainsi définie ? Quelle est son expression explicite ? Quelles propriétés a cette courbe ?

1.1 Introduction

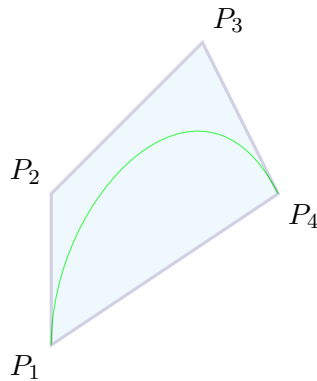
Les courbes de Bézier sont *juste* des courbes polynômiales. La base canonique pour les polynômes est la base des monômes. On peut interpréter géométriquement la signification des coefficients dans la base des monômes : les coefficients successifs représentent la valeur du polynôme en 0 et la valeur de ses dérivées successives en 0. C'est le développement de Taylor en 0 ! Cette information est donc très localisée.

La base de Bernstein est préférée aux bases de monômes pour son interprétation géométrique plus globale : les courbes de Bézier ont une expression de courbe polynômiale où le coefficient correspond à des points de contrôle qui caractérisent la forme de la courbe.

Approximation vs. interpolation Nous ne voulons pas interpoler des points, comme dans le chapitre précédent, mais seulement approximer un chemin défini de façon linéaire par morceaux. Pour cela, nous présentons 2 définitions des courbes de Bézier : une définition analytique, et une définition géométrique.

1.2 Courbe de Bézier : définition analytique

Maintenant, on ne veut pas interpoler tous les points mais juste suivre la forme des polygones de contrôle.



Le point de départ est un **ensemble de points de contrôle** $(P_0 \dots P_n)$ définissant un **polygone de contrôle**. On veut trouver une courbe ou fonction définie par ces points.

Définition analytique Soit un ensemble de points de contrôle. Alors la fonction P peut être définie par

$$\forall t, P(t) = \sum_{k=0}^n c_k(t) P_k$$

où $\sum_{k=0}^n c_k(t) = 1$.

Remarque : P est une combinaison $\sum c_k(t) = 1 \forall t$. La courbe est donc bien définie.

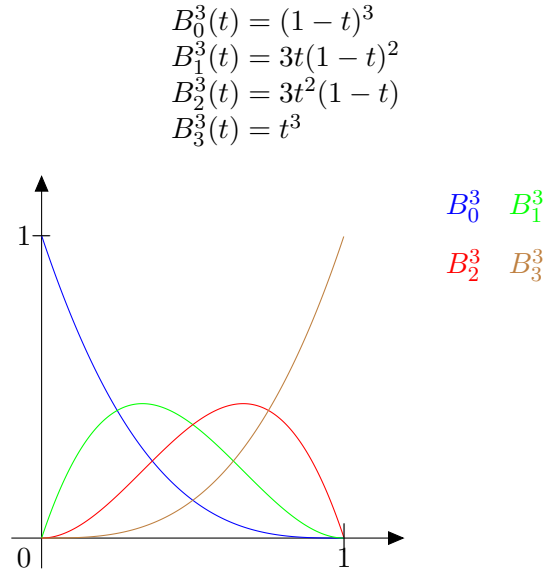
Si on veut que la fonction $P(t)$ suive le polygone de contrôle (créé en joignant tous les P_k), on a besoin de fonctions « bell-shaped » (en forme de cloche). Ces fonctions sont appelées **unimodales**. Ici, on a besoin de fonctions unimodales telles que le poids soit donné sur les points de contrôle tels que leur influence augmentera et diminuera. Un bon candidat est la base des polynômes de Bernstein sur $[0, 1]$.

Définition : on appelle **Polynômes de Bernstein** les polynômes, pour tout $t \in [0, 1]$, définis par :

$$B_k^n(t) = \binom{n}{k} (1-t)^{n-k} t^k$$

`bernstein(f,n,t)` : Bernstein polynomials (matlab)

Ecrivons et représentons les polynômes de Bernstein pour $n = 3$ sur $[0, 1]$:



Définition : on appelle **courbe de Bézier** la courbe définie par les points (P_0, \dots, P_n) dans la base des polynômes de Bernstein par :

$$P(t) = \sum_{k=0}^n B_k^n(t) P_k$$

On peut vérifier $\sum_{k=0}^n B_k^n(t) = 1 = [(1-t) + t]^n = \sum_{k=0}^n \binom{n}{k} (1-t)^{n-k} t^k$

Propriétés des courbes de Bézier : les polynômes de Bernstein apportent de nombreuses propriétés :

1. La courbe de Bézier interpole le premier et le dernier points de contrôle.
2. Une courbe de Bézier est tangente aux premier et dernier segments du polygone de contrôle.
3. Symétrie : la courbe définie par $(P_0 \dots P_n)$ est celle de $(P_n \dots P_0)$.
4. Elle est dans l'enveloppe convexe de ses points de contrôle.
5. Invariance affine : l'image d'une courbe de Bézier est une courbe de Bézier de l'image de ses points de contrôle.

Preuves

1. Evaluer la courbe de Bézier en 0 et 1. $P(0) = P_0, P(1) = P_n$
2. On peut calculer les dérivées d'une courbe de Bézier en 0 et 1 :
 $P'(0) = n(P_1 - P_0)$.
Plus généralement, la dérivée d'une courbe de Bézier est donnée par :

$$P'(t) = n \sum_{k=0}^{n-1} (P_{k+1} - P_k) B_k^{n-1}(t)$$

$$P(t) = \sum_{k=0}^n B_k^n(t) P_k = \sum_{k=0}^n \frac{n!}{k!(n-k)!} t^k (1-t)^{n-k}$$

Commençons par dériver les polynômes de Bernstein :

$$\begin{aligned} B_i^n(t)' &= \frac{i(n!)}{i!(n-i)!} t^{i-1} (1-t)^{n-i} - \frac{(n-i)n!}{i!(n-i)!} t^i (1-t)^{n-i-1} \\ &= n(B_{i-1}^{n-1}(t) - B_i^{n-1}(t)), \quad \forall t \in [0, 1] \end{aligned}$$

D'où la dérivée de la courbe de Bézier :

$$P'(t) = n \sum_{j=0}^n \left(B_{j-1}^{n-1}(t) - B_j^{n-1}(t) \right) P_j$$

Or $B_j^n(t) = 0, \forall j \notin \{0, \dots, n\}$ donc par transformation d'indice, on obtient, $\forall t \in [0, 1]$:

$$P'(t) = n \sum_{j=0}^{n-1} \underbrace{n(P_{j+1} - P_j)}_{\text{cf. hodographe}} B_j^{n-1}(t).$$

On remarque que les coefficients du polynôme dérivé P' correspondent aux différences entre deux points de contrôle successifs de P .

3. Symétrie : $B_i^n(t) = B_{n-i}^n(1-t)$.
4. Sur l'intervalle $[0, 1]$, les polynômes de Bernstein sont positifs (et leur somme est égale à 1) et donc chaque point de la courbe de Bézier est une combinaison convexe (de coefficients entre 0 et 1) des points de contrôle, et donc est dans l'enveloppe convexe des points de contrôle).
5. Une courbe de Bézier est une combinaison affine de points, et donc est préservée par transformation affine.

Propriété de la diminution de la variation Une dernière propriété est que la courbe n'oscille pas plus que son polygone de contrôle, ie, si on considère une droite, elle intersecte la courbe de Bézier au plus le nombre de fois qu'elle intersecte son polygone de contrôle.

Cette propriété n'est pas démontrée ici, mais pourra se déduire de la convergence de la subdivision... (voir section 1.5).

Approximation d'une courbe de Bézier

1. On a la propriété d'enveloppe convexe disant que la courbe de Bézier E a l'enveloppe convexe de ses points de contrôle. Le calcul de l'enveloppe convexe est en $O(n \log n)$ avec n le nombre de points de contrôle.
2. Une approximation plus grossière d'une courbe de Bézier est la suivante : Elle est incluse dans la boîte $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$ où x_{\min} (resp. y_{\min}) est l'abscisse (resp. ordonnée) minimale des points de contrôle et x_{\max} (resp. y_{\max}) leur abscisse (resp. ordonnée) maximale. Ce rectangle est appelé une **boîte englobante alignée sur les axes**.
3. Considérons la droite L joignant le premier et le dernier points de contrôle telle que $L(0) = P_0$ et $L(1) = P_n$. Si on veut approximer une courbe de Bézier par L , l'erreur maximale est bornée par la distance des points de contrôle à la ligne.

Formule de Wang

$$|P(t) - L(t)| \leq \frac{n(n-1)}{8} \|\Delta_2 P\|_\infty$$

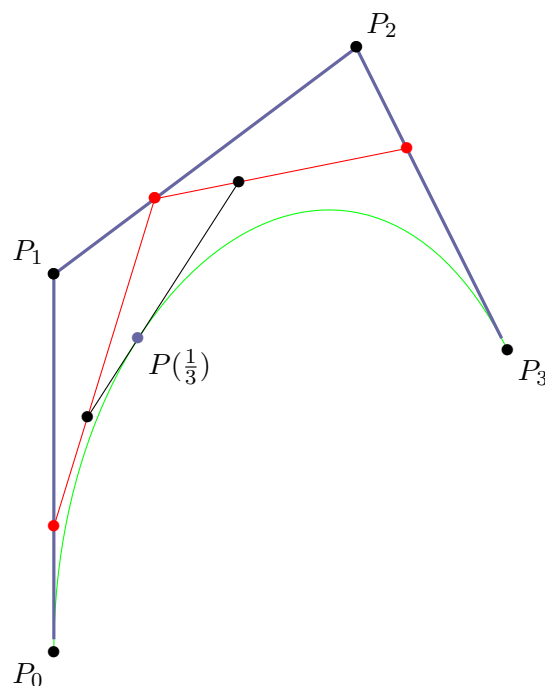
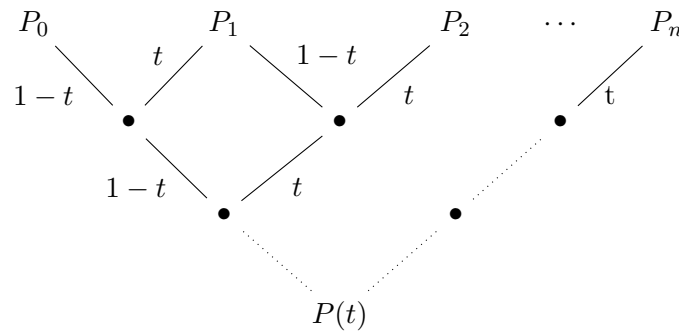
où $\|\Delta_2 P\| = |P_{k+2} - 2P_{k+1} + P_k|$ La différence entre les points de contrôle correspond aux points de contrôle de la dérivée seconde de P .

4. Nairn, Peters, Lutterkort (1999) et Reif (2001) donnent des bornes pour la distance entre une courbe de Bézier et son polygone de contrôle.
 - Paramétrisation
 - Fonction linéaire par morceaux l telle que les points de contrôle P_k et l'image de k/n . $\forall t \in [0, 1], \|P(t) - l(t)\| \leq N_\infty(d) \|\Delta_2 P\|_\infty$ où $N_\infty(n) = \frac{n}{8} - \frac{\omega}{8d}$ avec $\omega(n) = 0$ si n est pair et $\omega(n) = 1$ si n est impair.

1.3 Définition algorithmique

Une seconde façon d'introduire les courbes de Bézier est algorithmique, et correspond pour les courbes approximante, à l'algorithme de Neville pour les courbes interpolantes. Néanmoins, les coefficients considérés sont toujours les mêmes : on utilise juste des coordonnées barycentriques relatives aux paramètres 0 et 1.

Ce nouvel algorithme est l'**algorithme de De Casteljau**, un algorithme de programmation dynamique. Cet algorithme correspond aussi à la construction d'un point sur la courbe.



Exemple de l'algorithme De Casteljau évaluation pour $t = 1/3$

Interprétation : Notons que le chemin du point de contrôle P_k à la fin de l'algorithme c'est-à-dire $P(t)$ est réalisé en $n - k$ fois sur la droite et en k fois sur la gauche.

De plus, il y a $\binom{n}{k}$ chemins différents pour aller de P_k à $P(t)$ (parmi les n étapes, choisir k fois quand on part à gauche). Le coefficient associé à P_k est donc $\binom{n}{k}(1-t)^k t^k = B_k^n(t)$. On retrouve bien la base des polynômes de Bernstein associée aux points de contrôle P_k .

Comme l'algorithme de Neville, c'est un algorithme d'évaluation car ces algorithmes trouvent la valeur d'une fonction pour un paramètre donné à partir des points de contrôle $(P_i)_{i=1\dots n}$.

Complexité : en temps : $O(n^2)$. Si l'algorithme est récursif, alors complexité en $O(2^n)$, pas bon!!

en place : Cet algorithme peut être « implémenté sur place » en stockant les calculs directement dans la matrice des points de contrôle.

Remarque : Si on remplace P_k par 1 et tous les autres par 0 on obtient $B_k^n(t)$ voir le paragraphe suivant.

1.4 Floraison

Un outil adapté pour étudier Bézier sont **les floraisons** (Lyle Ramshaw 1985). Soit P_n un ensemble de polynômes de degré inférieur à n et soit \mathcal{F}_n l'ensemble des fonctions à n variables multi-affines.

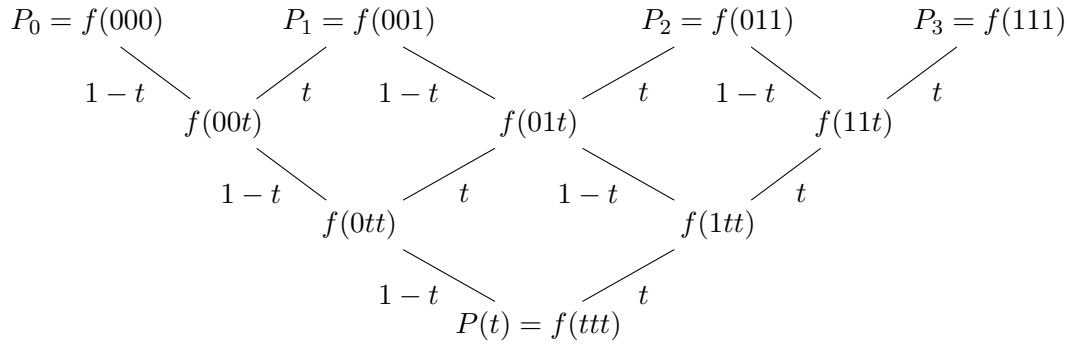
- Tous ces espaces sont des espaces vectoriels de dimension $n + 1$.
- Soit $f \in \mathcal{F}_n$ alors pour $x_1 \dots x_n \in \mathbb{R}$ on a $f((1 - \lambda)u + \lambda v, x_2 \dots x_n) = (1 - \lambda)f(u, x_2 \dots x_n) + \lambda f(v, x_2 \dots x_n)$ qui est affine sur la première variable.
- $f(x_{\sigma(1)} \dots x_{\sigma(n)}) = f(x_1 \dots x_n)$ pour une permutation σ arbitraire.
- La relation qui identifie une floraison f avec le polynôme correspondant P est la propriété diagonale $P(t) = f(t \dots t)$.

Propriété : Expression des points de contrôle de Bézier en termes de floraison.

Si P est un polynôme de degré inférieur à n , $(P_0 \dots P_n)$ sont des points de contrôle de Bézier et f sa floraison alors

$$P_i = f(\underbrace{0 \dots 0}_{n-i \text{ fois}}, \underbrace{1 \dots 1}_i).$$

Nous pouvons réinterpréter l'algorithme de De Casteljau en termes de floraison sur courbe de degré 3 :



Remarque : Une courbe de Bézier peut être définie sur un intervalle arbitraire $[a, b]$ par un changement affine de variable $P(u) = \sum_{i=0}^n P_i B_i^n \left(\frac{u-a}{b-a} \right)$.

Dans ce cas, on a une expression des points de contrôle par la floraison qui dépend de a et b :
 $P_i = f(\underbrace{a \dots a}_{n-i \text{ fois}}, \underbrace{b \dots b}_{i \text{ fois}})$.

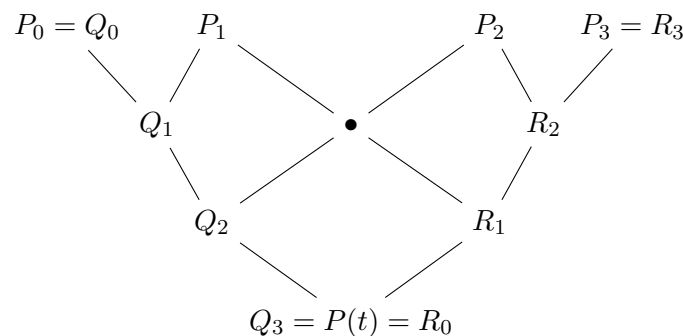
La notation de floraison sera aussi particulièrement utile pour les B-splines et leurs points de contrôle.

1.5 Subdivision

L'idée de subdivision est de diviser une courbe de Bézier en deux parties.

Soit une courbe de Bézier $P(t)$ de points de contrôle $(P_0 \dots P_n)$ sur $[0, 1]$, on considère le polynôme Q tel que $Q(t) = P(t/2)$. On remarque que P et Q paramétrisent la même courbe polynomiale, mais à des vitesses différentes. Si on se place sur l'intervalle $[0, 1]$ alors Q paramétrise la première moitié de P , en particulier car $Q(1) = P(1/2)$.

Q est une courbe polynomiale de même degré que P et doit donc avoir des points de contrôle $(Q_0 \dots Q_n)$ sur $[0, 1]$ qui seront des points de contrôle de P sur $[0, 1/2]$. Ces points apparaissent déjà dans l'algorithme de De Casteljau, on les trouve sur la gauche dans le diagramme de De Casteljau. De même les points de contrôle sur $[1/2, 1]$ sont $(R_0 \dots R_n)$ à droite.



Subdiviser sert à obtenir une meilleure approximation. On peut itérer ce processus : mes points de gauche $(Q_i)_{i=0 \dots 3}$ peuvent eux aussi être subdivisés à leur tour, ainsi que les points de droite, ce qui conduira à 4 familles de points de contrôle. Après i subdivisions, on aura 2^i familles de points de contrôle.

Propriété : Les polynômes successifs convergent vers la courbe.

Approximation avec subdivision Lorsque l'on subdivise, la taille de l'intervalle considéré est divisée par deux.

Corollaire : Si on applique la formule de Wang pour chaque famille de points $l_{01}^k(t)$ après i subdivisions, l'approximation linéaire $l_{01}(t)$ qui correspond à joindre les segments liant premiers et derniers points de contrôle de chaque famille a une différence avec le polynôme $P(t)$ majorée par :

$$\|l(t) - P(t)\| = \frac{n(n-1)}{8} \frac{1}{4^i} \max \|\Delta_2 P\|$$

\Rightarrow à utiliser pour l'intersection de courbes de Bézier.

1.6 Hodographe

On rappelle que la dérivée d'une courbe de Bézier P de degré n , de points de contrôle P_0, \dots, P_n est :

$$P'(t) = n \sum_{k=0}^{n-1} (P_{k+1} - P_k) B_k^n(t).$$

On note que $P'(t)$ représente géométriquement un vecteur (différence de points), et ses coefficients dans la base de Bernstein sont eux aussi des vecteurs.

Définition : On appelle l'**hodographe d'une courbe**, la courbe de Bézier des points $Q(t)$ telle que $\overrightarrow{OQ(t)} = P'(t)$ (voir la figure 1).

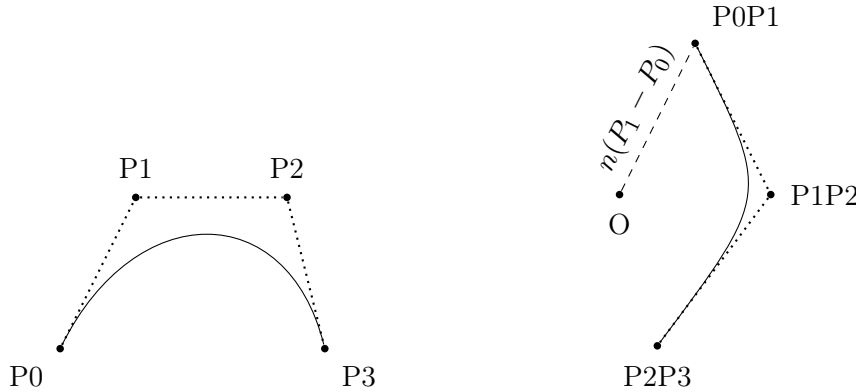


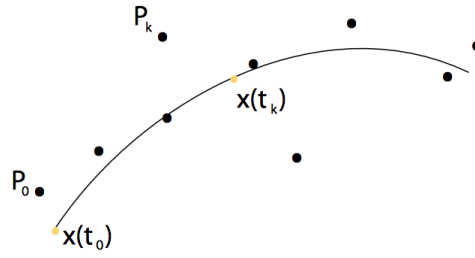
FIGURE 1 – Une courbe de Bézier (à gauche) et son hodographe (à droite).

Propriétés :

- la floraison de P' est $t \mapsto n(p(\underbrace{t, \dots, t}_{n-1 \text{ fois}}, 1) - p(\underbrace{t, \dots, t}_{n-1 \text{ fois}}, 0))$ où p est la floraison de P .
- Une condition nécessaire pour qu'une courbe de Bézier ait un point de rebroussement est que son hodographe (i.e. la courbe de son hodographe) passe par l'origine (dans ce cas la paramétrisation est irrégulière).
- La courbe dérivée ne change pas si une translation est appliquée à la courbe originale.

1.7 Approximation par moindres carrés

On considère un problème d'approximation d'un nuage de points par une courbe de Bézier. Soient $n+1$ points $P_0, \dots, P_n \in \mathbb{R}^2$ et $n+1$ paramètres correspondants $t_0 \leq t_i \leq t_n$ avec $t \in [a, b]$.



On cherche la courbe de Bézier $x(t) = \sum_{i=1}^n B_n^i(t)Q_i$, $t \in [a, b]$ qui approche le mieux les $n + 1$ points. On se ramène à un problème au sens des moindres carrés où on cherche les points de contrôles de la courbe de Bézier tels que :

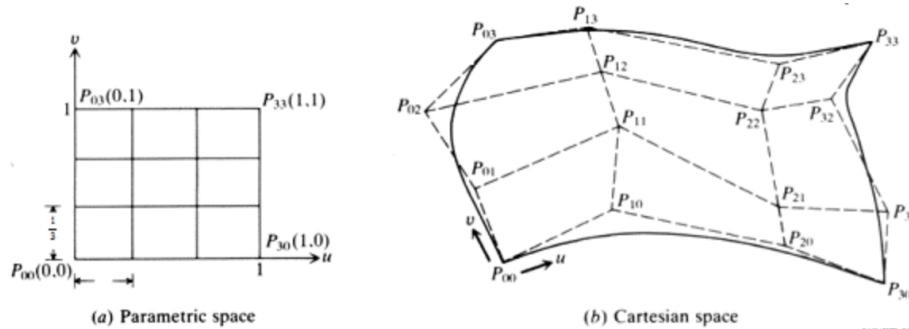
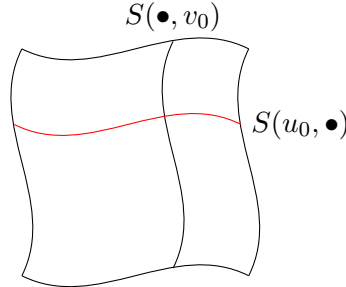
$$\min_Q \sum_{k=1}^n \|P_k - x(t_k)\|_2^2$$

1.8 Surfaces de Bézier

Surface paramétrique de Bézier :

$$S : \begin{array}{ccc} \mathbb{R}^2 & \rightarrow & \text{espace affine } \mathbb{R}^3 \\ (u, v) & \mapsto & S(u, v) \end{array}$$

On considère le domaine paramétrique suivant :



Il existe deux façon de calculer les surfaces de Bézier :

— **par produit tensoriel** : on considère une surface paramétrique S :

$$S : \mathbb{R}^2 \rightarrow \mathbb{R}^3, (u, v) \mapsto S(u, v)$$

On traite de courbes de contrôle $P_i(v)$ plutôt que de points de contrôle P_i .

$$S(u, v) = \sum_{j=0}^n \underbrace{\sum_{i=0}^m P_{ij} B_i^m(u) B_j^n(v)}_{P_j(u)}$$

Idée : considérer des courbes de contrôle plutôt que des points :

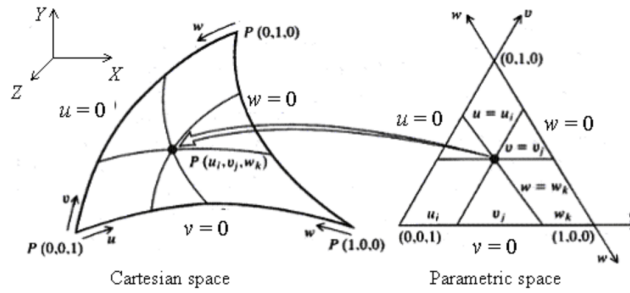
$$\forall j, P_j(u) = \sum_{i=0}^m P_{ij} B_i^m(u)$$

Calculer les surfaces de Bézier par produit tensoriel fonctionne bien. On travaille d'abord sur les lignes puis sur les colonnes : on travaille en **variables séparées**.

Ecriture matricielle des surfaces de Bézier :

$$S(u, v) = [B_n^0(u) \dots B_n^n(u)] \begin{bmatrix} P_{00} & \dots & P_{0m} \\ \vdots & & \vdots \\ P_{n0} & \dots & P_{nm} \end{bmatrix} \begin{bmatrix} B_m^0(v) \\ \vdots \\ B_m^m(v) \end{bmatrix}$$

— **par patch triangulaire** : les variables ici ne sont plus séparées.



En 1D : t varie sur un segment $[0,1]$: $t = 0(1-t) + 1t$

En 2D : étant donné que $u + v + w = 1$, (u, v) varie dans un triangle défini par les points $(0,0)$, $(0,1)$ et $(1,0)$ et on définit les coordonnées barycentriques par :

$$(u, v) = (1 - u - v)(0, 0) + u(1, 0) + v(0, 1).$$

$$S(u, v) = \sum_{\substack{i+j+k=n \\ i,j,k \geq 0}} P_{ijk} B_{ijk}^n(u, v)$$

avec $B_{ijk}^n(u, v) = \binom{n}{ijk} (1 - u - v)^k u^j v^i$

et $\binom{n}{ijk} = \frac{n!}{i!j!k!} = \frac{n!}{i!j!(n-i-j)!}$ où $k = n - i - j$.

On peut faire de la subdivision, mais les triangles obtenus par subdivision ont un mauvais aspect ratio.