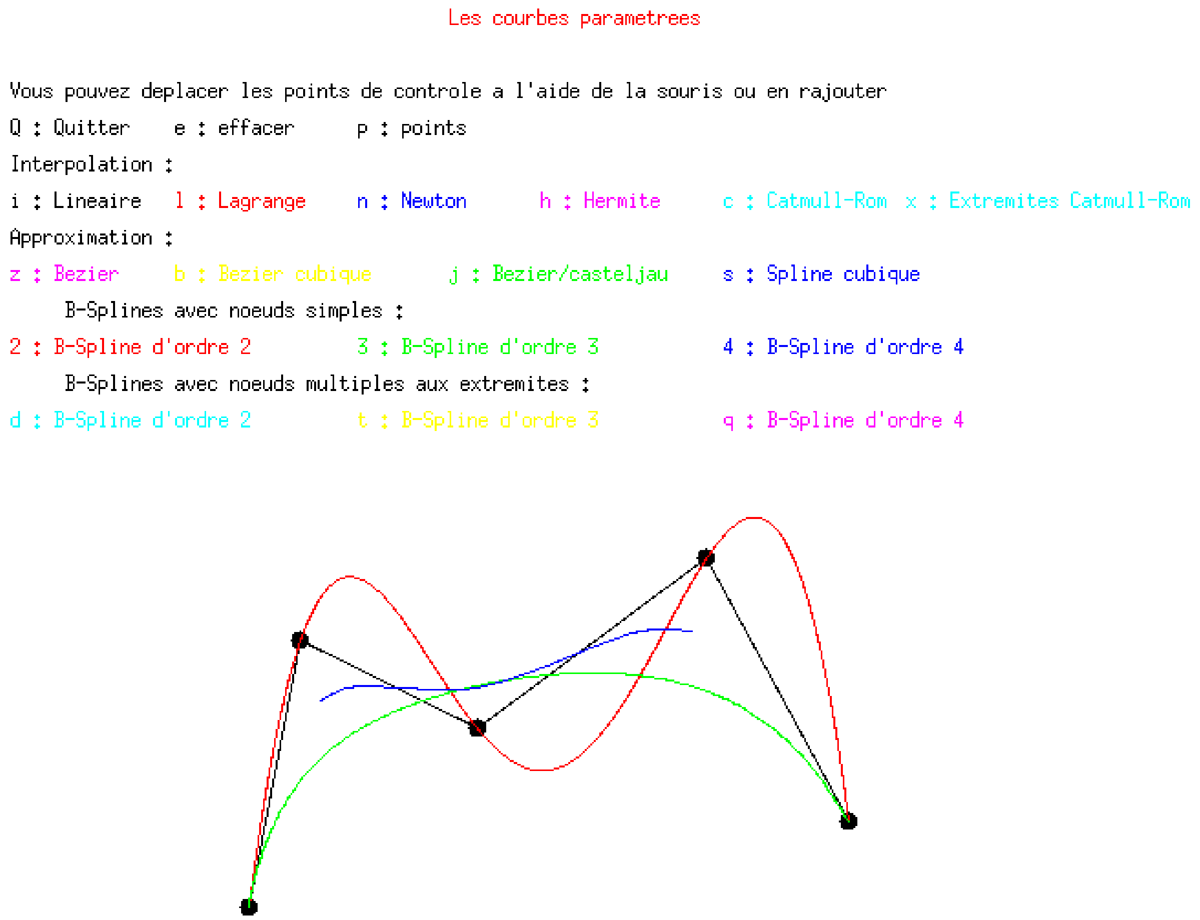
# Les courbes paramétrées

Céline Dubief



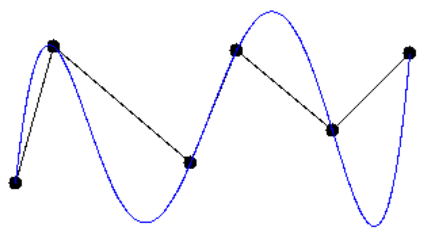

Université de Bourgogne Année 2008-2009

# Table des matières

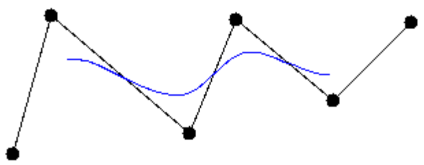
# Introduction

L'ajustement d'une courbe à un ensemble de points donnés est un problème important qui possède déjà plusieurs solutions . Il faut distinguer deux grandes classes à ce type de problèmes :

– si on désire trouver une courbe qui passe par un ensemble de points donnés, on a affaire à un problème d'interpolation.
– tandis que si on désire trouver une courbe qui passe près de cet ensemble de points, on a affaire à un problème d'approximation.



Interpolation



Approximation

Comparaison entre approximation et interpolation

L'ajustement d'une courbe ou d'une surface apparaît dans la création de projets (CAO : Conception Assistée par Ordinateur) ainsi que dans leur fabrication (FAO : Fabrication Assistée par Ordinateur).

Par exemple, la surface de la carroserie d'une voiture, ou encore d'une aile d'avion peut-être spécifiée par un ensemble fini de points déterminés par calcul, par expérimentation ou tout simplement par des considérations artistiques.

# 1 Les courbes paramétrées

## 1.1 Définition

On appelle courbe paramétrée dans l'espace $\mathbb{R}^2$ toute application continue :

$$\left\{ \begin{array}{cccc} Q: & [a,b] & \rightarrow & \mathbb{R}^2 \\ & t & \longmapsto & Q(t) = (x(t), y(t)) \end{array} \right.$$

Pour simplifier, on se limitera aux courbes en 2D (points de coordonnées $(x(t), y(t))$). Cependant, les courbes à points de contrôle dans le plan s'étendent à des courbes dans l'espace en prenant des points de contrôle en dimension 3.

Pour que cette application soit continue, il faut que les 2 fonctions, $t \longmapsto x(t)$ et $t \longmapsto y(t)$, qui sont des fonctions de $\mathbb{R}$ dans $\mathbb{R}$, soient continues.

Lorsque $Q$ est à valeur dans $\mathbb{R}^2$, la courbe $Q$ est appelée courbe plane.

## 1.2 Courbes de degré d quelconque

Une courbe $Q : [a,b] \rightarrow \mathbb{R}^2$ qui associe $Q(t) = (x(t), y(t))$ est dite polynomiale si chacune des coordonnées $x(t), y(t)$, s'exprime comme un polynôme en $t$.

Si $Q$ est polynomiale, il existe un nombre $d$, appelé degré de la courbe $Q$, et des coefficients $a_i, b_i$ pour $i = 0, \ldots, d$ tels que pour tout $t \in [a,b]$ on ait l'égalité 1 suivante :

$$Q(t) = \left\{ \begin{array}{ccc} x(t) & = & \sum_{i=0}^{d} a_i t^i \\ y(t) & = & \sum_{i=0}^{d} b_i t^i \end{array} \right. \tag{1}$$

Soit $T$ le vecteur à $d+1$ composantes $T = (t^d, t^{d-1}, \ldots, t, 1)$
Soit $C$ la matrice

$$C = \begin{pmatrix} a_d & b_d \\ \vdots & \vdots \\ a_0 & b_0 \end{pmatrix}$$

L'équation (1) est équivalente à l'écriture matricielle suivante :

$$(1) : Q(t) = (x(t), y(t)) = T.C$$

## 1.3 Les courbes paramétriques cubiques

Une courbe cubique est une courbe polynomiale ayant pour degré $d \leq 3$.
L'équation (1) s'écrit :

$$(1) : \left\{ \begin{array}{ccc} x(t) & = & a_3 t^3 + a_2 t^2 + a_1 t + a_0 \\ y(t) & = & b_3 t^3 + b_2 t^2 + b_1 t + b_0 \end{array} \right. \tag{2}$$

Pour obtenir des segments de courbes, on considère que $t$ appartient à un intervalle $[min, max]$ (fréquemment $[0,1]$).

Soit le vecteur $T = (t^3, t^2, t, 1)$, l'équation (2) s'écrit sous la forme $Q(t) = (x(t), y(t)) = T.C$ avec $C = \begin{pmatrix} a_3 & b_3 \\ a_2 & b_2 \\ a_1 & b_1 \\ a_0 & b_0 \end{pmatrix}$ et $t \in \mathbb{R}$. Ainsi, connaissant la matrice $C$ et une valeur de $t$, on peut calculer le point $Q(t)$ à partir de l'équation matricielle précédente.

Quatre coefficients numériques constants sont nécessaires pour définir chacune des deux équations paramétriques. On impose alors quatre contraintes mathématiques pour définir l'équation selon chacun des deux axes.

Pour les courbes cubiques, il est fréquent de décomposer $C$ en deux matrices $M$ et $G$ :

- $M$ est une matrice $4*4$ (appelée matrice de base), matrice constante qui dépend du type de courbes considérées (courbes hermitiennes, courbes de Bézier, courbes splines..)
- $G$ est appelé matrice géométrique qui dépend des points de contrôle et caractérise les quatre contraintes géométriques sur la courbe.

$$\begin{aligned} Q(t) &= (x(t), y(t)) = T.C = T.M.G \\ &= (t^3\ t^2\ t\ 1) \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{pmatrix} \begin{pmatrix} g_{1x} & g_{1y} \\ g_{2x} & g_{2y} \\ g_{3x} & g_{3y} \\ g_{4x} & g_{4y} \end{pmatrix} \end{aligned}$$

La matrice $G$ permet de représenter les 4 contraintes géométriques s'appliquant à la courbe. Il s'agit habituellement de la position de 4 points dans l'espace de représentation. La courbe générée évoluera "entre" ces quatre points. Mais il peut aussi s'agir des dérivées de la courbe aux points de contrôle, ou encore de contraintes de continuité de raccordement avec d'autres courbes dans le cas de courbes polynomiales par morceaux.

La matrice $M$ permet d'attribuer des "poids" respectifs à chacun des coefficients $t^i$ s'appliquant à une contrainte géométrique, permettant ainsi de définir la "forme" de la courbe.

De bons choix de matrice permettront d'obtenir des tracés de courbe approximant la ligne polygonale à 4 sommets définie par les quatre points présentant différentes caractéristiques.

# 2 Interpolation

Soient $n+1$ points, $P_i$, $i = 0...n$. On souhaite trouver la courbe $Q(t)$ passant par ces points.

## 2.1 Interpolation linéaire

### 2.1.1 Définition

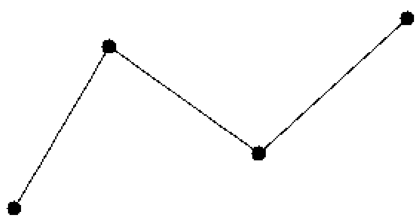
La méthode la plus simple est de relier chaque points par des segments de droite.

Pour interpoler, on a besoin de deux points et aussi d'un paramètre $t$ entre 0 et 1, celui-ci permet de dire où l'on se trouve sur le segment qui relie ces deux points. Si $t = 0$, alors on se trouve sur le premier point, si $t = 1$ alors notre position sera sur le dernier point. La formule d'interpolation linéaire est simplement :

$$Q(t) = (1-t)P_0 + tP_1 \tag{3}$$

### 2.1.2 Propriétés

Il s'agit alors simplement de joindre les paires de points $(P_i, P_{i+1})$ de façon indépendante.



Interpolation linéaire

Dans beaucoup de cas l'interpolation linéaire n'est pas satisfaisante parce que la fonction que l'on désire interpoler est beaucoup trop éloignée d'une droite.

## 2.2 Interpolation de Lagrange

### 2.2.1 Définition

Étant donné $n+1$ points $(x_0, y_0), (x_1, y_1), ..., (x_n, y_n)$ (avec $(x_i)_{0 \leq i \leq n}$ distinct 2 à 2). On souhaite trouver une unique courbe passant par tous ces points.

Les $(x_i)_{0 \leq i \leq n}$ sont appelés points d'interpolation. Les $(y_i)_{0 \leq i \leq n}$ représentent les valeurs d'interpolation.

Les polynômes de Lagrange associés à ces points sont les polynômes définis par :

$$l_j(x) = \prod_{i=0, i \neq j}^{n} \frac{x - x_i}{x_j - x_i} = \frac{x - x_0}{x_j - x_0} \cdots \frac{x - x_{j-1}}{x_j - x_{j-1}} \frac{x - x_{j+1}}{x_j - x_{j+1}} \cdots \frac{x - x_n}{x_j - x_n}$$

On a en particulier deux propriétés :

1. $l_j$ est de degré $n$ pour tout $j$
2. $l_i(x_j) = \delta_{ij}$, $0 \leq i, j \leq n$, c'est-à-dire $l_i(x_j) = \begin{cases} 1 & si \quad i = j \\ 0 & si \quad i \neq j \end{cases}$

Le polynôme défini par

$$L(x) = \sum_{j=0}^{n} y_j l_j(x) \tag{4}$$

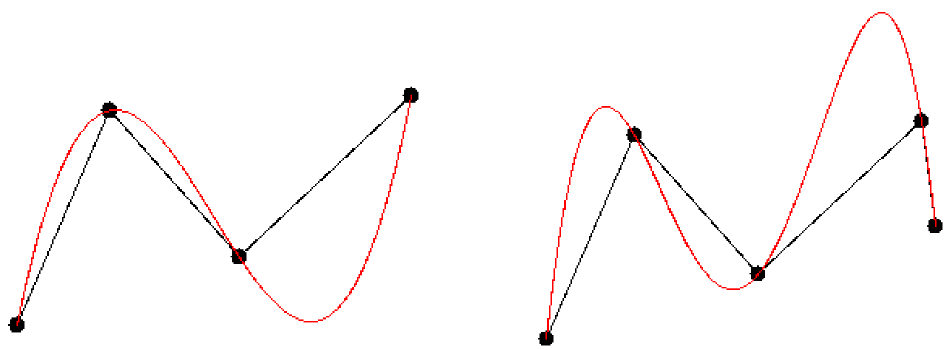
est l'unique polynôme de degré au plus $n$ vérifiant $L(x_i) = y_i$ pour tout $i$.

### 2.2.2 Propriétés

L'interpolation de Lagrange donne des fonctions de degré $n$. On observe alors une oscillation lorsqu'on élève le degré en ajoutant un point.

Modifier un point de contrôle affecte toute la courbe (non local). Si on désire supprimer ou ajouter un point à l'ensemble des données, il faudra recommencer tous les calculs.

Le résulat est une courbe trop sinueuse.



Interpolation de Lagrange

De tels polynômes ne sont pas pratiques puisque du point de vue numérique, il est difficile de déduire $l_{k+1}$ de $l_k$. Pour cela, on introduit le polynôme d'interpolation de Newton.

Remarque sur le programme :
Les abscisses des points donnés doivent être obligatoirement dans l'ordre croissant. (Le programme les tri dans l'ordre croissant avant d'interpoler.)

## 2.3 Interpolation de Newton

### 2.3.1 Définition

Les polynômes $e_k$ de la base de Newton sont définis comme suit :

$$e_k(x) = \prod_{i=0}^{k-1}(x - x_i) = (x - x_0)(x - x_1)\dots(x - x_{k-1}), k = 1, ..., n.$$

avec pour convention

$$e_0 = 1$$

ainsi

$$\begin{array}{rcl} e_1 & = & (x - x_0) \\ e_2 & = & (x - x_0)(x - x_1) \\ \vdots & & \vdots \\ e_n & = & (x - x_0)(x - x_1)\cdots(x - x_{n-1}) \end{array}$$

L'ensemble des polynômes $(e_k)_{0 \leq k \leq n}$(base de Newton) forment une base de l'espace $P_n$ des polynômes de degré au plus $n$, puisqu'il s'agit d'une famille de $n+1$ polynômes de degré échelonné.

Le polynôme d'interpolation de Newton de degré $n$ relatif à la subdivision $\{(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)\}$ s'écrit :

$$\begin{array}{rcl} P_n(x) & = & \displaystyle\sum_{k=0}^{n} \alpha_k e_k(x) \\ & = & \alpha_0 + \alpha_1(x - x_0) + \alpha_2(x - x_0)(x - x_1) + \dots \\ & & +\alpha_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}) \end{array}$$

avec

$$P_n(x_i) = y_i, \forall i = 0, \dots, n.$$

Il faut alors déterminer les coefficients $(\alpha_k)_{0 \leq k \leq n}$. Pour cela, on va utiliser les différences divisées.

### 2.3.2 Différences divisées

Le polynôme d'interpolation de Newton de degré $n$ , $P_n(x)$ évalué en $x_0$ donne :

$$P_n(x_0) = \sum_{k=0}^{n} \alpha_k e_k(x_0) = \alpha_0 = y_0 = [y_0]$$

De manière générale, on note

$$[y_i] = y_i, \quad \forall i = 0, \ldots, n$$

$[y_0]$ est appelée différence divisée d'ordre 0.

Le polynôme d'interpolation de Newton de degré $n$, $P_n(x)$ évalué en $x_1$ donne :

$$\begin{aligned} P_n(x_1) &= \sum_{k=0}^{n} \alpha_k e_k(x_1) = \alpha_0 + \alpha_1(x_1 - x_0) \\ &= [y_0] + \alpha_1(x_1 - x_0) = [y_1] \end{aligned}$$

d'où :

$$\alpha_1 = \frac{[y_1] - [y_0]}{x_1 - x_0} = [y_0, y_1]$$

$[y_0, y_1]$ est appelée différence divisée d'ordre 1.

On obtient alors par récurrence :

$$\alpha_k = \frac{[y_1, \ldots, y_k] - [y_0, \ldots, y_{k-1}]}{y_k - y_0} = [y_0, \ldots, y_k]$$

$[y_0, \ldots, y_k]$ est alors appelée différence divisée d'ordre $k$.

En pratique lorsqu'on veut déterminer par exemple la différence divisée d'ordre 3 , on a besoin des quantités suivantes :

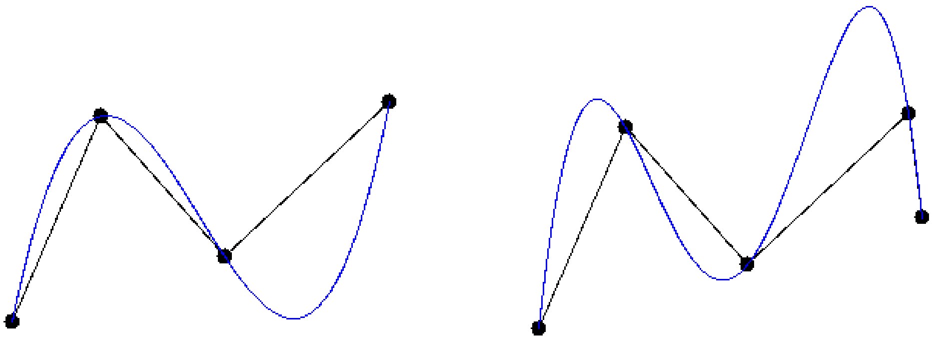
$$\begin{bmatrix} x_0 & [y_0] & & & \\ x_1 & [y_1] & [y_0, y_1] & & \\ x_2 & [y_2] & [y_1, y_2] & [y_0, y_1, y_2] & \\ x_3 & [y_3] & [y_2, y_3] & [y_1, y_2, y_3] & [y_0, y_1, y_2, y_3] \end{bmatrix}$$

Le polynôme d'interpolation de Newton de degré $n$ s'écrit donc à l'aide des différences divisées successives :

$$P_n(x) = [y_0] + \sum_{k=1}^{n} [y_0, \ldots, y_k] e_k(x) \tag{5}$$

### 2.3.3 Propriétés

Les contraintes des polynômes de Lagrange et Newton étant les mêmes, ceux-ci fournissent un résultat identique en tout point x. Seule la forme du polynôme change.



Interpolation de Newton

Les interpolations de Lagrange ou de Newton, qui fournissent facilement un polynôme prenant des valeurs données, présente l'inconvénient dans certains cas de donner une qualité médiocre. Entre les points d'interpolation la différence entre une fonction et son polynôme d'interpolation peut être grande, même si le nombre de points tend vers l'infini. Pour remédier à cela on peut essayer d'utiliser non seulement les valeurs d'une fonction mais aussi celles de ses dérivées, comme c'est le cas dans l'interpolation de Hermite.

## 2.4 Interpolation de Hermite

### 2.4.1 Définition

Soient $P_0 = (x_0, y_0)$ et $P_2 = (x_2, y_2)$ deux points de contrôle. Soient $P'_0 = (x'_0, y'_0)$ et $P'_2 = (x'_2, y'_2)$ deux vecteurs de $\mathbb{R}^2$. La courbe hermitienne avec $P_0$ et $P_2$ pour points de contrôle, et $P'_0$ et $P'_2$ pour dérivées aux points de contrôle est l'unique courbe cubique $Q : [0,1] \to \mathbb{R}^2$ telle que :

$$\left\{ \begin{array}{l} Q(0) = P_0 \ et \ Q(1) = P_2 \\ Q'(0) = P'_0 \ et \ Q'(1) = P'_2 \end{array} \right. \tag{6}$$

La forme algébrique d'une courbe de Hermite est donnée par :

$$Q(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

où t est une variable réelle appartenant à [0,1]. Avec les conditions (6), on obtient le système suivant :

$$\left\{ \begin{array}{rcc} P(0) & = & a_0 \\ P'(0) & = & a_1 \\ P(1) & = & a_3 + a_2 + a_1 + a_0 \\ P'(1) & = & 3a_3 + 2a_2 + a_1 \end{array} \right. \Leftrightarrow \left\{ \begin{array}{rcc} a_3 & = & 2P_0 + P'_0 - 2P_2 + P'_2 \\ a_2 & = & -3P_0 - 2P'_0 + 3P_2 - P'_2 \\ a_1 & = & P'_0 \\ a_0 & = & P_0 \end{array} \right.$$

Ainsi

$$Q(t) = (2P_0 + P'_0 - 2P_2 + P'_2)t^3 + (-3P_0 - 2P'_0 + 3P_2 - P'_2)t^2 + P'_0 t + P_0$$

C'est-à-dire

$$Q(t) \;=\; (t^3\ t^2\ t\ 1)\begin{pmatrix} 2 & 1 & -2 & 1 \\ -3 & -2 & 3 & -1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} P_0 \\ P'_0 \\ P_2 \\ P'_2 \end{pmatrix} = TM'_H G'$$

Ici, pour simplifier le programme,(dans notre programme nous ne connaissons que les positions des points) nous fixerons les tangentes en fonction d'autres points :

$$P'_0 = P_1 - P_0 \; et \; P'_2 = P_3 - P_2$$

D'où

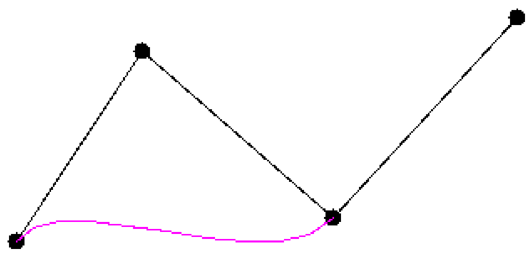
$$\begin{aligned} Q(t) &= TM'_H G' = TM'_H \begin{pmatrix} P_0 \\ P_1 - P_0 \\ P_2 \\ P_3 - P_2 \end{pmatrix} \\ &= (t^3\ t^2\ t\ 1)\begin{pmatrix} 2 & 1 & -2 & 1 \\ -3 & -2 & 3 & -1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix}\begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix} \\ &= (t^3\ t^2\ t\ 1)\begin{pmatrix} 1 & 1 & -3 & 1 \\ -1 & -2 & 4 & -1 \\ -1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix} = TM_H G \end{aligned}$$

Donc, la courbe de Hermite est donnée par :

$$Q(t) = TM_H G \tag{7}$$

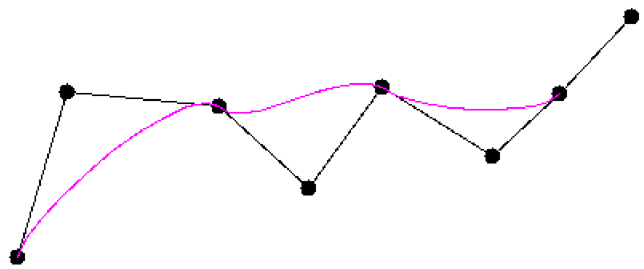
### 2.4.2 Propriétés

Pour construire une interpolation de Hermite, il faut donc spécifier la position et le vecteur tangent à chaque point.



Interpolation de Hermite cubique

Etant donnée une liste de points et de tangentes, on peut construire une cubique par morceaux qui passe en tout point, en calculant la cubique de Hermite pour chaque morceau. La courbe obtenue est alors de continuité $C^1$.



Interpolation de Hermite par morceaux

## 2.5 Interpolation de Catmull-Rom

### 2.5.1 Définition

Les courbes de Catmull-Rom peuvent être vues comme des courbes de Hermite composées, dans lesquelles les tangentes aux points de contrôles internes sont générées automatiquement à l'aide d'une procédure géométrique relativement simple.
Pour chaque point intérieur $P_i$, la tangente $P_i'$ en ce point est donnée par :

$$P_i' = \frac{P_{i+1} - P_{i-1}}{2}$$

Or, la courbe de Hermite $Q(t)$ est donnée par :

$$Q(t) \quad = \quad TM_H'G' = (t^3\ t^2\ t\ 1) \begin{pmatrix} 2 & 1 & -2 & 1 \\ -3 & -2 & 3 & -1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} P_0 \\ P_0' \\ P_2 \\ P_2' \end{pmatrix}$$

Ainsi, la courbe de Catmull-Rom sera :

$$\begin{aligned}
Q(t) &= TM'_H G'_i = TM'_H \begin{pmatrix} P_i \\ \frac{P_{i+1}-P_{i-1}}{2} \\ P_{i+1} \\ \frac{P_{i+2}-P_i}{2} \end{pmatrix} \\
&= (t^3\ t^2\ t\ 1) \begin{pmatrix} 2 & 1 & -2 & 1 \\ -3 & -2 & 3 & -1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -\frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix} \\
&= (t^3\ t^2\ t\ 1) \frac{1}{2} \begin{pmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{pmatrix} \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix} = TM_C G_i
\end{aligned}$$

Donc, la courbe de Catmull-Rom est donnée par :

$$Q(t) = TM_C G_i \tag{8}$$

où $T = \big(t^3\ t^2\ t\ 1\big), M_C = \frac{1}{2} \begin{pmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{pmatrix}$

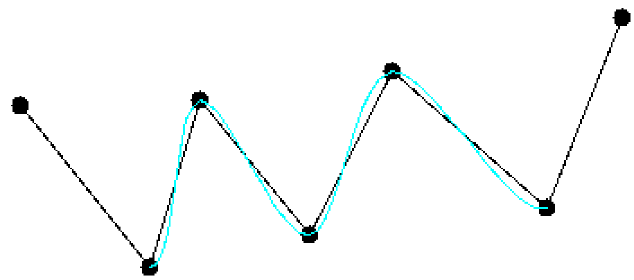
et $G_i = \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix}, 1 \leq i \leq n-2.$,

### 2.5.2 Propriétés

Les calculs permettant d'obtenir les vecteurs tangents intérieurs sont très simples.

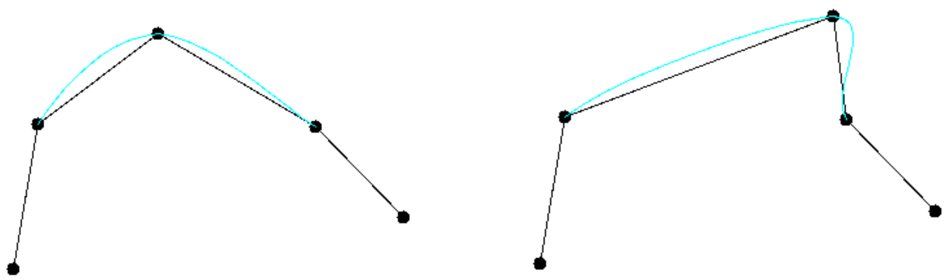
La courbe de Catmull-Rom passe par tous les sommets de la ligne brisée, à l'exception du premier et du dernier, commence au deuxième point et finit à l'avant dernier.

Pour le vecteur géométrique $G_i$ la courbe commence pour $t = 0$ en $P_i$, et finit pour $t = 1$ en $P_{i+1}$. Tous les morceaux se rejoignent en leurs extrémités pour $t \in [0,1]$. Ces extrémités sont les points de contrôle. La courbe obtenue est alors de continuité $C^1$.



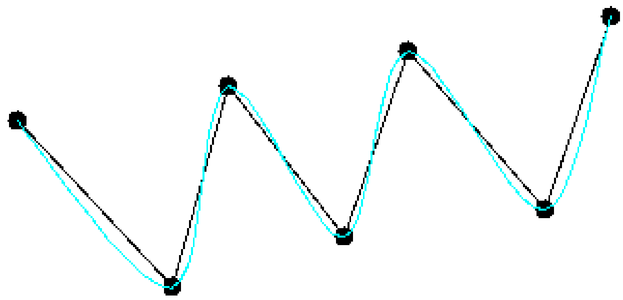
Interpolation de Catmull-Rom

Un vecteur tangent interne ne dépend pas de la position du point interne.La courbe est tangente au point $P_i$ à la direction donnée par le segment $P_{i-1}$, $P_{i+1}$.



Interpolation de Catmull-Rom

Une solution simple pour passer par le premier et le dernier point consiste à doubler ces points.



Interpolation de Catmull-Rom en doublant les extrémités

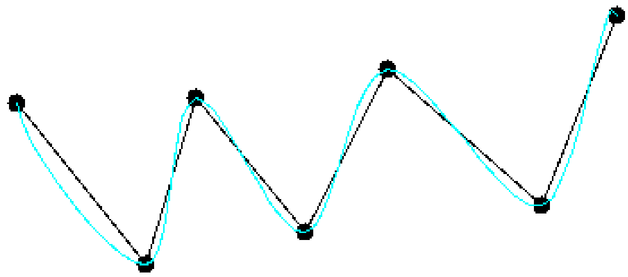
Une autre solution, consiste,à fixer les vecteurs tangents aux 2 extrémités :

– le vecteur tangent à $P_0$ est $\frac{(P_1P_0)-(P_2P_1)}{2}$, d'où

$$M_C = \frac{1}{2}\begin{pmatrix} 2 & -2 & 0 & 0 \\ -3 & 2 & 1 & 0 \\ -1 & 2 & -1 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix}$$

– le vecteur tangent à $P_{N-1}$ est $\frac{(P_{N-1}P_{N-2})-(P_{N-2}P_{N-3})}{2}$, d'où

$$M_C = \frac{1}{2}\begin{pmatrix} 1 & -3 & 5 & -3 \\ -1 & 4 & -7 & 4 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & 2 & 0 \end{pmatrix}$$



Interpolation de Catmull-Rom avec conditions aux extrémités

# 3 Approximation

Soient $n+1$ points, $P_i$, $i = 0...n$. On souhaite trouver la courbe $Q(t)$ passant près de ces points.

## 3.1 Les courbes de Bezier

### 3.1.1 Définition

Si $P_0, P_1, ..., P_n$ sont les $n+1$ points de contrôle. La courbe de Bézier définie par ces points est :

$$C(t) = \sum_{i=0}^{n} B_i^n(t) P_i,\ t \in [0,1] \tag{9}$$

où

$$B_i^n = C_i^n t^i (1-t)^{n-i} P_i,\ t \in [0,1]$$

est la base de Bernstein, avec $C_i^n$ représentant les coefficients binomiaux

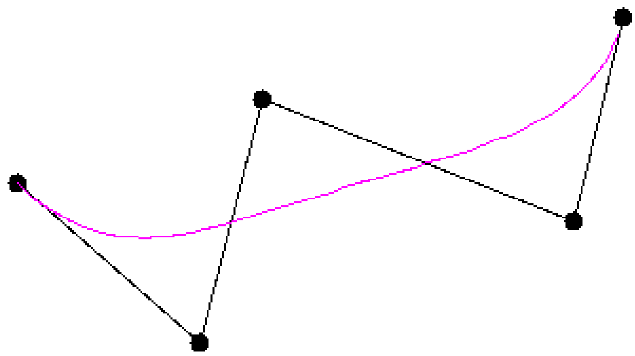
$$C_i^n = \frac{n!}{(n-i)!i!},\ i \in [0,n]$$

Pour calculer la courbe de Bézier, il suffit de faire varier le paramètre $t$ de 0 à 1.

### 3.1.2 Propriètés

La courbe de Bezier commence par le premier, fini par le dernier point et ne passe pas obligatoirement par les autres points.

La courbe est tangente aux côtés du polygone en ses points extrêmes. C'est-à-dire, elle est tangente au segment $[P_0P_1]$ en $P_0$ et au segment $[P_nP_{n-1}]$ en $P_n$.



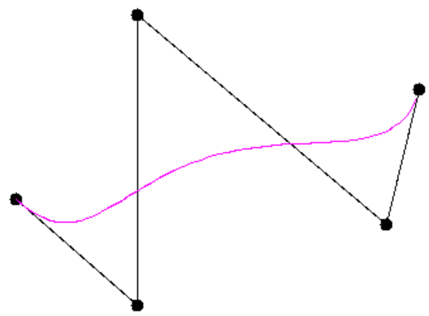
Courbe de Bézier

Les courbes de Bézier sont invariantes par translation, changement d'échelle et rotation.

Une courbe de Bézier est toujours contenue à l'intérieur de l'enveloppe convexe déterminée par l'ensemble des points de contrôle.

De plus, la courbe est intégralement modifiée quand on modifie la position d'un seul point de contrôle, ce qui entraîne sa réévaluation intégrale. Il n'y a donc pas de contrôle local sur la forme de la courbe.

Chaque point de contrôle exerce une attraction sur la portion de courbe qui se trouve près de lui. Cette attraction est proportionnelle au coefficient $B$ correspondant.



Courbe de Bézier

Le calcul des $C_i^n$ n'est pas économique en temps, de même pour les mises à la puissance en $t$ et en $1-t$.

Le degré de la courbe dépend du nombre de points de contrôle. Si on a $n+1$ points de contrôle, la courbe est de degré $n$.

## 3.2 Les courbes de Bézier cubiques

Il y a plusieurs façon d'approcher les courbes de Bézier. L'une d'elles pour les courbes de Bézier cubiques est de dire qu'elle est similaire à la formulation de Hermite, à l'exception du fait qu'au lieu d'utiliser les vecteurs tangentes, la formulation de Bézier utilise des points de contrôle auxiliaires pour définir les vecteurs tangents.

Si $n=3$ (4 points de contrôles), la courbe $C(t)$ peut-être présentée sous la forme d'une courbe paramétrique cubique définie par les 4 points de contrôle. Les 4 contraintes géométriques $G_1$, $G_2$, $G_3$ et $G_4$ sont alors les positions de ces 4 points de contrôle.
On a alors :

$$\begin{aligned}
C(t) &= (x(t),y(t)) = \sum_{i=0}^{3} B_i^n(t)P_i \\
&= C_0^3 t^0(1-t)^3 P_0 + C_1^3 t^1(1-t)^2 P_1 + C_2^3 t^2(1-t)^1 P_2 + C_3^3 t^3(1-t)^0 P_3 \\
&= (1-3t+3t^2-t^3)P_0 + 3(t-2t^2+t^3)P_1 + 3(t^2-t^3)P_2 + t^3 P_3 \\
&= \begin{pmatrix} t^3 t^2 t 1 \end{pmatrix} \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} P_{0x} & P_{0y} \\ P_{1x} & P_{1y} \\ P_{2x} & P_{2y} \\ P_{3x} & P_{3y} \end{pmatrix}
\end{aligned}$$

La matrice de base utilisée est donc :

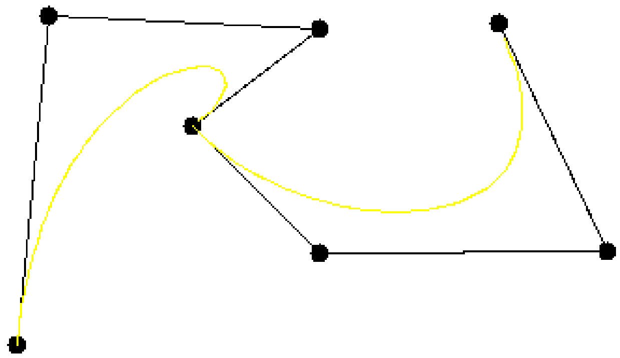
$$M_B = \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Remarques :
La courbe de Bézier de points de contrôle $P_0$, $P_1$, $P_2$ et $P_3$ est la courbe de Hermitienne d'extrémités $P_0$ et $P_3$ avec pour dérivée aux extrémités $T_0$ et $T_3$ :

$$T_0 = 3(P_1 - P_0) \; et \; T_3 = 3(P_3 - P_2)$$

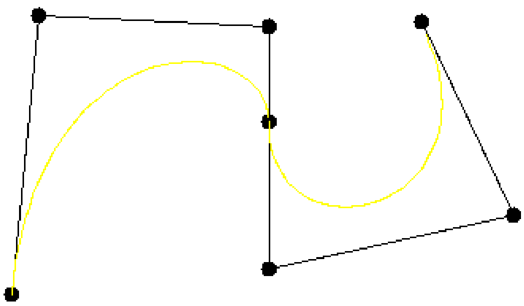
### 3.2.1 Propriétés

Pour relier deux courbes de Bézier cubiques entre elles, il faut que le dernier sommet définissant la première courbe soit aussi le premier sommet de la seconde, on a alors une courbe $C^0$.



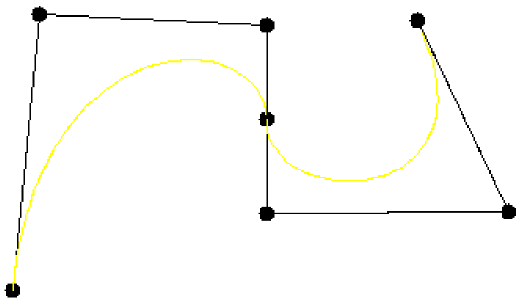
Courbe de Bézier cubique de classe $C^0$

Si on veut obtenir une courbe $C^1$, il faut que le dernier côté du premier polygone et le premier côté du second polygone soient colinéaires.



Courbe de Bézier cubique de classe $C^1$

Enfin, si on souhaite une obtenir une courbe $C^2$, il faut que le vecteur $P_0P_1$ de la seconde courbe soit identique avec le vecteur $P_{n-1}P_n$ de la première courbe.



Courbe de Bézier cubique de classe $C^2$

## 3.3 Les courbes de Bézier avec l'algorithme de De Casteljau

Le calcul d'une courbe de Bézier n'est pas des plus simples. Ainsi pour nous simplifier la tâche, nous allons utiliser l'algorithme de De Casteljau.

### 3.3.1 Définition

Dans le cas cubique :

Considérons la ligne brisée ayant pour sommets $P_0$, $P_1$, $P_2$ et $P_3$, et une valeur $t \in [0,1]$. On peut construire, pour $i = 0,1,2$, un point

$$P_i^{(1)}(t) = (1-t)P_i + tP_{i+1}$$

Pour $i = 0,1,2$ le point $P_i^{(1)}(t)$ se trouve sur le segment $[P_i, P_{i+1}]$.

Considérons maintenant la ligne brisée formée par les trois points $P_0^{(1)}(t)$, $P_1^{(1)}(t)$ et $P_2^{(1)}(t)$. On peut appliquer le même procédé en posant pour $i = 0,1$ :

$$P_i^{(2)}(t) = (1-t)P_i^{(1)} + tP_{i+1}^{(1)}$$

Enfin, on pose :

$$Q(t) = P_0^{(3)} = (1-t)P_0^{(2)} + tP_0^{(2)}$$

Alors la courbe $Q : [0,1] \mapsto \mathbb{R}^2$ obtenue est la courbe de Bézier cubique ayant $P_0$, $P_1$, $P_2$ et $P_3$ pour points de contrôle.

Généralisation :

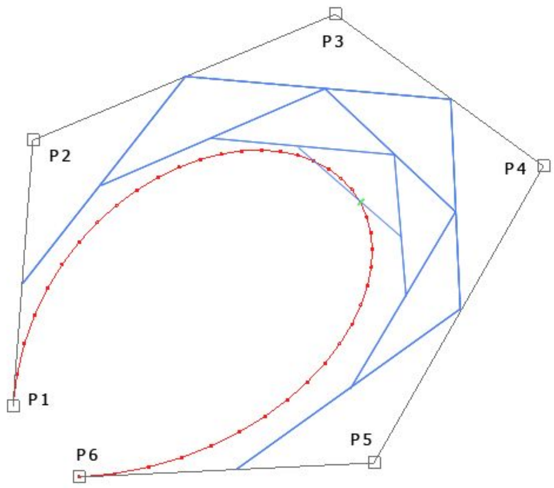
En généralisant l'algorithme de De Casteljau présenté dans le cas cubique on peut généraliser les courbes de Bézier à un nombre $n \geq 2$ quelconque de points de contrôle. Nous définissons ainsi, une courbe de Bézier d'ordre $n$.

Soient $P_0, \dots P_n$ les points de contrôle, et $t \in [0,1]$. Pour définir le point $Q(t)$, nous procédons de la façon suivante :

1. On pose $b_i^{(0)} = P_i$ pour $i = 0, \dots, n$
2. Pour $r = 1, \dots, n$, et $i = 0, \dots, n-r$, on pose :

$$b_i^{(r)}(t) = (1-t)b_i^{(r-1)}(t) + tb_{i+1}^{(r-1)}(t)$$

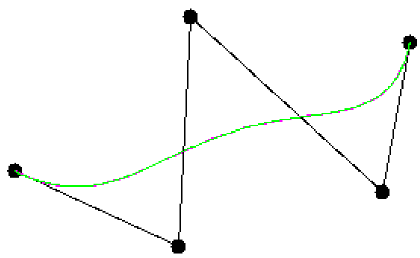
On définit ainsi par récurrence un point $Q(t) = b_0^{(n)}(t)$.




Courbe de Bézier avec l'algorithme de De Casteljau

### 3.3.2 Propriétés

On retrouve les même propriètés que dans la partie *Les courbes de Bézier*.



Courbe de Bézier avec l'algorithme de De Casteljau

## 3.4 Les courbes B-Splines non rationnelles uniformes

La formulation des courbes B-Splines est semblable à celle des courbes de Bézier. La différence fondamentale réside dans la formulation des fonctions mélanges. Les courbes de Bézier s'expriment comme une combinaison linéaire des polynômes de Bernstein, alors que les courbes B-splines se définissent à partir d'une base de fonctions.

### 3.4.1 Définition

Soient $P_0,\ldots,P_n$ des points de contrôle. La courbe B-Spline $C_k$ d'ordre $k$ et de degré $k-1$ ayant $T=(t_0,\ldots,t_{n+k})$ pour vecteur nodal et les $P_i$ pour points de contrôle s'exprime comme :

$$C_k(t)=\sum_{i=0}^{n}P_iN_{i,k}(t) \tag{10}$$

Où les $N_{i,k}(t)$ sont les fonctions mélanges définies par les formules suivantes :

$$\begin{aligned} N_{i,1}(t) &= \begin{cases} 1 & si\ t_i \le t < t_{i+1} \\ 0 & partout\ ailleurs \end{cases} \\ N_{i,k}(t) &= w_{i,k}(t)N_{i,k-1}(t)+(1-w_{i+1,k}(t))N_{i+1,k-1}(t) \end{aligned}$$

avec

$$w_{i,k}(t)=\begin{cases} 0 & si\ t_{i+k-1}=t_i \\ \frac{t-t_i}{t_{i+k-1}-t_i} & sinon \end{cases}$$

La courbe $C_k$ s'exprime donc comme une somme des fonctions $N_{i,k}$ pondérée par les points de contrôle.

Notons que, outre les points de contrôle, la courbe dépend des nombres $t_i$, que l'on peut essayer de bien choisir. Un choix classique est de prendre les $t_i$ en progression arithmétique, c'est-à-dire que l'on prend des intervalles $[t_i,t_{i+1}]$ de longueur indépendante de $i$. On parle alors de courbe B-Spline uniforme. Les valeurs $t_i$ sont les éléments d'un vecteur appelé vecteur noeud. Un vecteur noeud est tout simplement une suite d'entiers positif $x_i$ tels que $x_i \le x_{i+1}$ pour tout $i$. On peut par exemple choisir :

- un vecteur noeud simple : Par exemple, pour $i=0,\ldots,n+k$, on pose $t_i=i$. Les valeurs des $t_i$ indiquent l'intervalle dans lequel peut varier le paramètre $t$. Contrairement aux courbes de Bézier où $t$ variait toujours dans l'intervalle $[0,1]$. Avec les B-Splines, le paramètre $t$ peut varier dans l'intervalle $[t_{k-1},t_{n+1}]$. La courbe $C_k$ est alors formée de $n-k+2$ morceaux de courbes.
- un vecteur noeud multiple au bord :
  En design paramétrique on préfère travailler avec des noeuds de multiplicité $k$ aux deux extrémités de l'intervalle, c'est-à-dire $t_0=\ldots=t_k$ et $t_{n+1}=\ldots=t_{n+k}$. L'ordre $k$ de la courbe se trouve alors reflétée dans le vecteur noeud parce qu'il faut spécifier des noeuds de multiplicité $k$ aux deux extrémités du vecteur. Par

exemple, si l'on considère un polygone de 5 sommets alors une courbe B-Spline d'ordre $k = 3$ utilisera le vecteur de noeud suivant $T = (0,0,0,1,2,3,3,3)$.
Les valeurs des $t_i$ indiquent également l'intervalle dans lequel peut varier le paramètre $t$. Avec ce vecteur noeud, le paramètre $t$ peut varier dans l'intervalle $[0, t_{max}]$ où $t_{max}$ est la valeur maximale des éléments du vecteur noeud. Dans le cas où il n'y a aucun sommets multiples, la valeur de $t_{max}$ est $n-k+1$.
Le vecteur noeud peut se calculer comme suit :

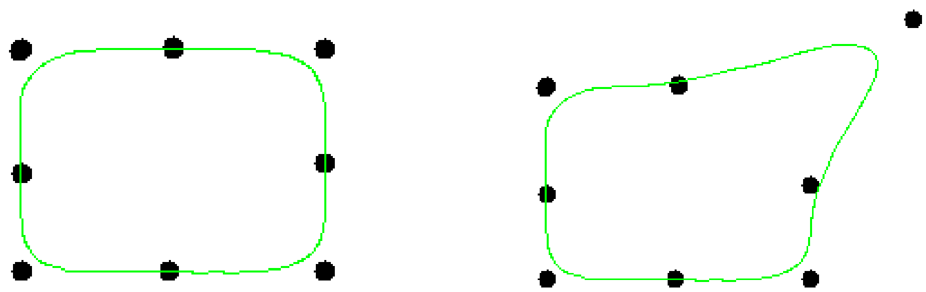
$$\begin{cases} x_i &= 0 & si & i < k \\ x_i &= i-k+1 & si & j \leq i \leq n \\ x_i &= n-k+2 & si & i > n \end{cases}$$

La courbe passe ainsi par $P_0$ et $P_n$ et est tangente au premier et dernier segment du polygone de contrôle. Cela permet un meilleur contrôle de la courbe.
L'ordre $k$ de la courbe est toujours un entier tel que $2 \leq k \leq n+1$.

### 3.4.2 Propriétés

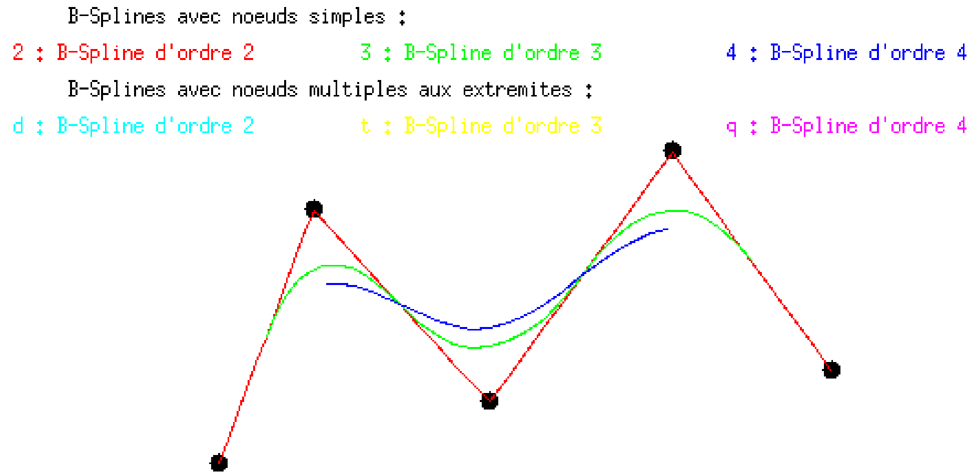
Les courbes B-Splines possèdent la propriète d'enveloppe convexe, c'est-à-dire que tous les points d'une courbe B-Spline se trouvent dans l'enveloppe convexe des points de contrôle. De plus, les courbes B-Splines sont invariantes par translation, changement d'échelle et rotation.

Les courbes B-Splines sont à contrôle local, ce qui est un grand avantage par rapport aux courbes de Bézier. Chaque sommet affecte la forme de la courbe dans un intervalle donné. En effet, la fonction $B_{i,k}$ s'annule en dehors de l'intervalle $[t_i, t_{i+k+1}[$. Donc si l'on modifie un point de contrôle $P_i$, le point $C_k(t) = \sum_i P_i N_{i,k}(t)$ n'est modifié que pour $t \in [t_i, t_{i+k+1}[$ (seule une partie de la courbe est modifiée). De plus, pour tout $t \in [t_j, t_{j+1}[$ , le point $C_k(t)$ ne dépend que des points de contrôle $P_i$ tels que $t \in [t_i, t_{i+k+1}]$ , c'est-à-dire des points $P_i$ tels que $i \leq j \leq i+k$. Le point $C_k(t)$ ne dépend donc que des points de contrôle $P_{j-k}, \ldots, P_j$, soit k+1 points de contrôle. L'avantage de cette propriété est que si l'on modifie un seul point de contrôle, seule la partie modifiée aura à être recalculée, d'où un gain de temps et de mémoire.

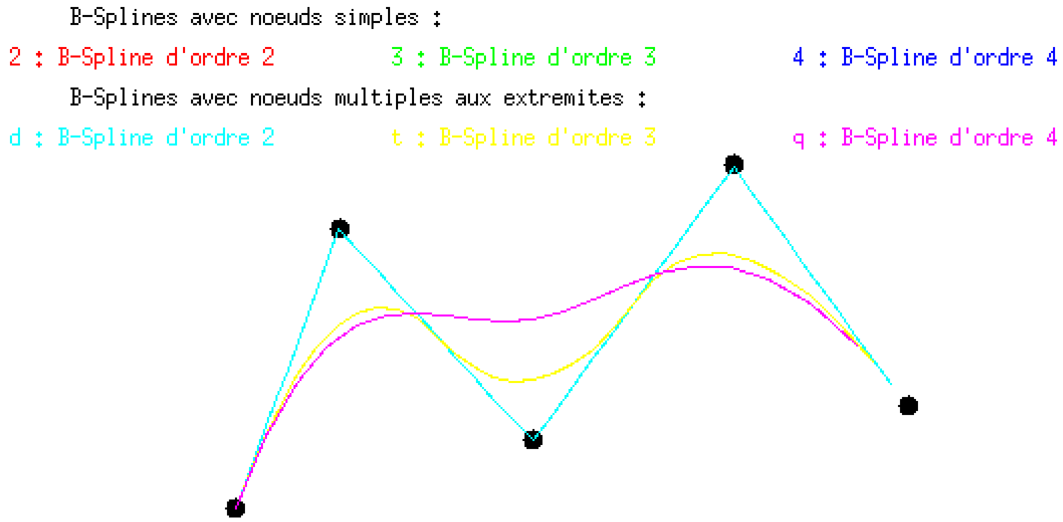


Courbes B-Splines d'ordre (contrôle local)

La base de B-Spline permet de changer l'ordre de la courbe sans changer le nombre de sommets du polygone. Cela est très intéressant car une courbe d'ordre élevé est plus difficile à contrôler et à calculer avec précision. Plus le degré de la courbe B-Spline est petit, plus la courbe se rapproche du polygone. Inversement, plus le degré de la courbe est grand, plus la courbe est "‘tendue"’ entre les points extrêmes.



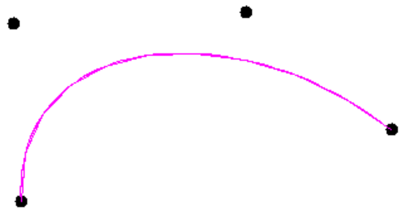

Courbes B-Splines d'ordre 2, 3 et 4 avec un vecteur noeud simple




Courbes B-Splines d'ordre 2, 3 et 4 avec un vecteur noeud multiple aux extrémités

Dans le cas du vecteur noeud multiple au bord, si l'ordre de la courbe est égale le nombre de sommet du polygone, alors la courbe B-Spline obtenue est identique à la courbe de Bézier correspondante.

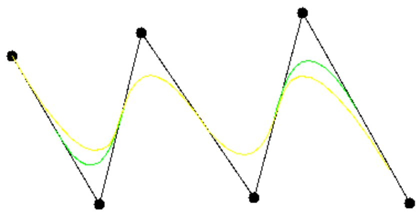
B-Spline d’ordre 3



B-Spline d’ordre 4

Courbes B-Splines d’ordre 2 et 3 correspondant aux courbes de Bezier

Une courbe B-Spline d'ordre 3 est toujours tangente aux milieux des cotés du polygone.



B-Spline d'ordre 3 (jaune : noeud simple, vert : noeud multiple)

## 3.5 Les courbes B-Splines cubiques uniformes non rationnelles

Dans cette section, on se place dans le cas des B-Splines cubiques uniformes non rationnelles, c'est-à-dire :

- de degré 3 (cubiques), k=4 ;
- le vecteur nodal est défini par $t_i = i$ (uniformes) ;
- les coordonnées des points de la courbe sont polynomiales par morceaux (non rationnelles).

Pour calculer la courbe B-Spline $C_k$, on peut utiliser l'algorithme de Boor :
On fixe un vecteur noeud $(t_0, ..., t_{n+k})$ et un polygone de contrôle $P_0, ..., P_n$. Soit $t \in [t_i, t_{i+1}[$.
On pose $P_j^0 = P_j$ pour $i - k \leq j \leq i$.
Puis, pour $r = 0, ..., k-1$, on pose $P_j^{r+1} = w_{j,k-r}(t)P_j^r + (1 - w_{j,k-r}(t))P_{j-1}^r$ pour $i - k + r + 1 \leq j \leq i$.
Alors $P_i^k = C_k(t)$.

Nous utiliserons cet algorithme pour calculer les B-Splines cubiques.

Soient $P_0, ..., P_n$ des points de contrôle. Soit $t_i = i$ pour $i = 0, ..., n+k$. La courbe B-Spline cubique $C_k$ d'ordre $k$ est polynomiale de degré 3 sur chaque intervalle $[t_i, t_{i+1}]$, avec $i = 3, ..., n$. Pour tout $t \in [t_i, t_{i+1}]$, le point $C_k(t)$ ne dépend que des points de contrôle $P_{i-3}, P_{i-2}, P_{i-1}, P_i$, soient 4 points de contrôle. De plus, la courbe $C_k$ dépend linéairement des points de contrôle.

Par conséquent, sur chaque intervalle $[t_i, t_{i+1}]$, avec $i = 3, ..., n$, la courbe $C_k$ s'exprime par une équation matricielle :

$$C_k(t) = T.M_{BS}.G_i$$

où $T = ((t - t_i)^3, (t - t_i)^2, (t - t_i), 1)$, $M_{BS}$ est une matrice constante (qui ne dépend pas de $i$ en raison de l'invariance $B_{i,k}(t+1) = B_{i-1,k}(t)$ due à la relation d'uniformité $t_i = i$), et la matrice $G_i$ est la matrice composée des coordonnées des points de contrôle dont dépend la courbe sur l'intervalle considéré :

$$G_i = \begin{pmatrix} P_{i-3} \\ P_{i-2} \\ P_{i-1} \\ P_i \end{pmatrix}$$

Détermination de la matrice :

On se propose maintenant de déterminer la matrice $M_{BS}$. Pour cela, notons

$$M_{BS} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

Détermination de la première colonne de $M_{BS}$ :

Le point $C_k(t)$ pour $n = 4$ avec $P_0 = (1,0)$ $P_1 = P_2 = P_3 = (0,0)$. Le point $C_k(t) = (x(t), y(t))$ est défini pour $t \in [3,4]$. La coordonnée $y$ de chacun des points de contrôle est nulle. Calculons donc $x(t)$. On a l'initialisation par les coordonnées $x$ des points de contrôle :

$$P_0^0 = 1, P_1^0 = P_2^0 = P_3^0 = 0$$

Pour $k = 1$, on a par la relation de récurrence définissant $P_i^k$ :

$$P_0^1 = \frac{4-t}{3}, P_1^1 = P_2^1 = 0$$

pour $k = 2$ :

$$P_0^2 = \frac{(4-t)^2}{6}, P_2^1 = 0$$

pour $k = 3$ :

$$x(t) = P_0^3 = \frac{1}{6}(-(t-t_3)^3 + 3(t-t_3)^2 - 3(t-t_3) + 1)$$

D'autre part , la relation $X_k(t) = T.M_{BS}.G_i$ donne :

$$x(t) = a_{11}(t-t_3)^3 + a_{21}(t-t_3)^2 + a_{31}(t-t_3) + a_{41}$$

En identifiant les coefficients des deux polynômes égaux, on obtient :

$$a_{11} = -\frac{1}{6}, a_{21} = 3\frac{1}{6}, a_{31} = -3\frac{1}{6}, a_{41} = \frac{1}{6}.$$

Détermination des autres vecteurs colonnes :

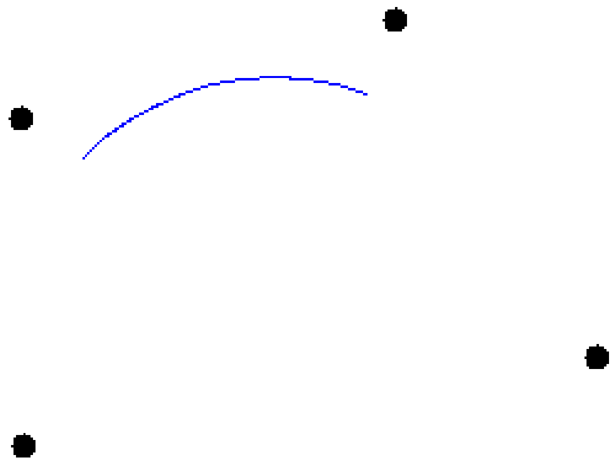
On procède comme la première colonne :
- pour la deuxième colonne, on prend $P_1 = (1,0)$, $P_0 = P_2 = P_3 = (0,0)$ ;
- pour la troisième colonne, on prend $P_2 = (1,0)$, $P_0 = P_1 = P_3 = (0,0)$ ;
- pour la quatrième colonne, on prend $P_3 = (1,0)$, $P_0 = P_1 = P_2 = (0,0)$.
On vérifie alors que

$$M_{BS} = \frac{1}{6}\begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix}$$

### 3.5.1 Propriétés

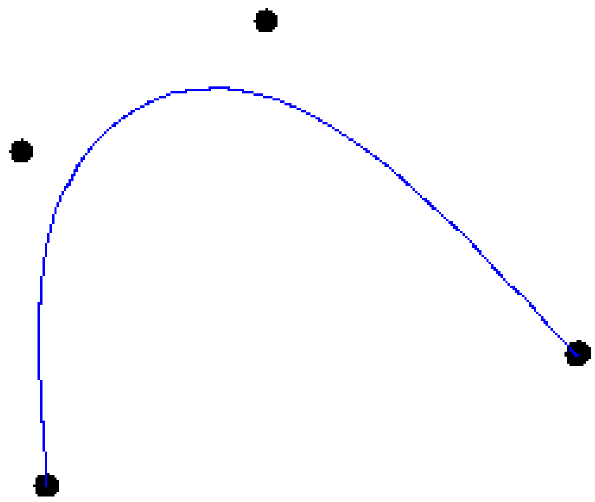
Les courbes B-Splines cubiques sont de classe $C^2$.

Les B-Splines cubiques ne passent pas par les sommets de la ligne brisée, y compris le premier et le dernier.



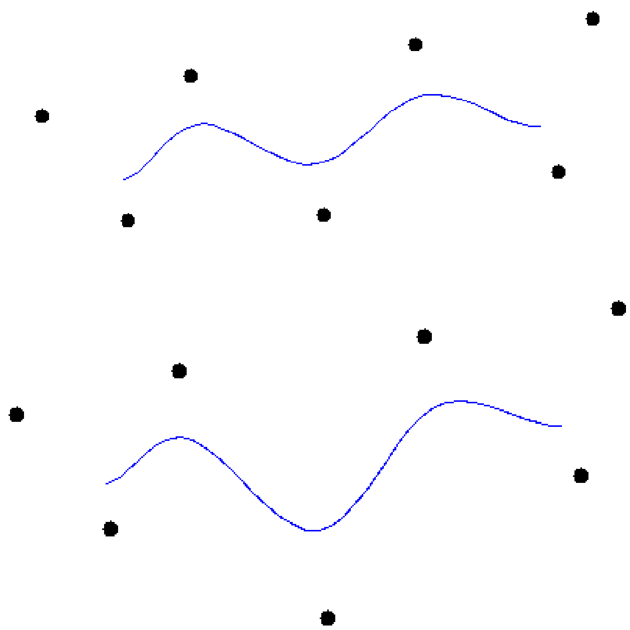
Courbe de B-Spline cubique

Une solution simple pour passer par le premier et le dernier point consiste à tripler ces points.



Courbe de B-Spline cubique

Un point de contrôle ne peut modifier que le segment qui suit et celui qui précède.



Courbes de B-Spline cubique

# Conclusion

Dans ce dossier, on a observé les différentes façons d'ajuster une courbe à un ensemble de points. En effet, on peut soit vouloir interpoler les points, soit les approximer.

Dans le premier cas, on peut utiliser les interpolations de Lagrange ou de Newton, mais les courbes obtenues sont généralement loin du résultat attendu. On peut aussi utiliser des courbes paramétriques cubiques, comme les courbes de Hermite ou encore de Catmull-Rom.

Dans le deuxième cas, on utilise généralement les courbes de Bézier ou les courbes B-Splines. A la différence des courbes de Bézier, les courbes B-splines ne sont pas polynomiales, mais polynomiales par morceaux. L'avantage des courbes B-Splines uniformes non rationnelles est qu'elles permettent de lisser un polygone au moyen d'une courbe paramétrique cubique telle que la modification de la position d'un seul point de contrôle ne modifie qu'une seule partie de la courbe (caractère local). De plus, elles sont de degré peu élevé, ce qui permet des calculs plus précis et moins coûteux.