

# Cours Approximation Géométrique

Géraldine Morin, Sandrine Mouysset

14 avril 2017

Ce cours s'organise en CTD, TPs et projet (voir Section pour plus de détail), avant chaque CTD, un exercice est à préparer (en rose dans le texte), et pour chaque partie, les motivations sont données, et les compétences à acquérir (en bleu) doivent permettre à un étudiant d'évaluer si il maîtrise les notions nécessaires.

## Intérêt et motivation du cours

### 0.1 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 chaun vient avec une trace écrite de ses réflexions.

**La question** On vous demande de développer (dans l'entreprise de rêve qui vient de vous embaucher) un petit logiciel qui permettrait à des artistes de créer des objets 3D, et de le modifier de façon interactive. Quelle représentation géométrique utiliseriez vous pour modéliser ces objets ? Quelle structure de données et méthodes utiliseriez vous pour implémenter le modèle choisi ?

### 0.2 Comment modéliser des objets 3D ? Des courbes ? Des surfaces ?

Bilan/brain storming/mise en commun....

- forme implicite  $\begin{cases} x^2 + y^2 = 1 \\ x^2 + y^2 + z^2 - R^2 = 1 \end{cases}$   
2 équations dont l'intersection donne une équation courbe.
- Courbes paramétriques  $\begin{cases} x = x(t) \\ y = y(t) \\ z = z(t) \end{cases}$
- Maillage issu d'une discrétisation d'EDP
- Surfaces paramétriques  $\begin{cases} x = x(s, t) \\ y = y(s, t) \end{cases}$  courbes paramétrées à 2 indices.

Plusieurs difficultés peuvent apparaître :

- Si un point  $M \in$  surface ?  
Avec forme implicite, OK.  
Courbe paramétrique, pas OK.
- Génération de points d'une surface  
Forme implicite, pas OK  
Maillage ou courbes paramétriques, OK

### 0.3 Motivation

Les contextes dans lesquels sont utilisés ces modèles sont :

- Les logiciels de CAO (vidéo modélisation CAO) ;
- Les logiciels d'animation (vidéo Tony deRose).

## 0.4 Organisation et travail demandé

La classe va se dérouler 'en classe' avec :

- 3 CTD : Interpolation fonctionnelle (Philippe Berger)
- 5 CTD : Courbes paramétriques, Approximation, Subdivision, cas des surfaces.
- 9 TP : 3 TP sur les courbes (2 séances par TP), 1 projet sur les surfaces.
- Evaluation : 1 note Projet, 1 note Examen (comptent chacune 50% de la matière)

Votre travail :

- Quantité : 20-25h de travail en dehors des séances (processus de Bologne) ;
- comprendre et si besoin rattravailler les CTDs (vieu examen en ligne)
- terminer les TP si nécessaire en dehors des séances
- Faire un projet :
  - Passer au cas des surfaces ;
  - Appliquer les concepts vu en classe, en explorant un nouveau modèle (avantages de ce modèle, applications visées, limitations) ;
  - Implémentation de ce modèle géométrique ;
  - Illustration sur des exemples bien choisis
  - création d'une vidéo d'illustration de votre travail.

## 0.5 Compétences et savoir faire

A la fin de cet enseignement, un(e) étudiant(e) doit

- connaître les modèles utilisés pour la modélisation paramétrique des courbes et surfaces ;
- connaître les avantages des modèles paramétriques en modélisation 3D ;
- savoir choisir un modèle paramétrique particulier pour une application visée ;
- savoir implémenter un modèle de surface paramétrique particulier.

Auto-évaluation : Vous retrouvez en début de chaque séance, en bleu, la liste de toutes les compétences à acquérir durant ce cours (ces compétences sont données sous moodle pour que vous puissiez auto-évaluer vos acquis).

Suite à ce cours, vous devez répondre "oui" à ces questions :

- je sais définir un espace affine ;
- je connais les éléments d'un espace affine, et je sais quelles opérations s'appliquent sur ces éléments ;
- j'interprète géométriquement les opérations sur l'espace affine ;
- je sais définir une base de l'espace affine, et je sais aussi définir un repère de cet espace ;
- je sais comparer les espaces vectoriels et les espaces affines ;
- je connais les limitations de la modélisation affine, et je sais que pour certaines transformations, il faut se tourner vers des espaces plus généraux : les espaces projectifs.
- Trouver les coordonnées barycentriques d'un point d'un espace affine dans une base affine de cet espace ;

- Etant données des coordonnées barycentriques d'un point sur une famille affine, dessiner le point dans l'espace affine ;
  - caractériser une transformation affine, algébriquement et géométriquement.
- 
- Etant donnés des points  $(x_i, y_i)$   $i = 0 \dots n$  je sais définir une courbe  $y = f(x)$  polynomiale de degré  $n$  qui passe par ces points.
  - je sais que cette courbe est unique, et je l'exprime comme une combinaison linéaire des polynômes de Lagrange ;
  - Je sais définir les polynômes de Lagrange relatifs à une famille de points, et montrer qu'il forment une base de l'espace des polynômes de degré inférieur ou égal à  $n$ .
- 
- je sais définir une courbe paramétrique polynomiale de degré  $n$  passant pas des points  $(x_i, y_i)$   $i = 0 \dots n$  ;
  - Je sais que la courbe ainsi définie dépend des temps de passage choisis  $t_i$  pour chaque point à interpoler ;
  - Je sais construire le polynôme interpolant, itérativement à partir de l'algorithme de Neville.
- 
- je sais définir une surface paramétrique polynomiale de bi-degré  $n$  passant pas des points  $(x_{i,j}, y_{i,j})$   $i, j = 0 \dots n$  explicitement ;
  - Je sais que la surface ainsi définie dépend de temps d'interpolation (je dois savoir combien, et comment les définir et les utiliser pour définir les fonctions de base) ;
  - Je sais aussi construire cette surface interpolante itérativement en utilisant l'algorithme de Neville.
- 
- je sais construire une courbe de Bézier ;
  - je connais la base polynomiale correspondant aux courbes de Bézier ;
  - je connais 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 linéaire par morceaux proposée ;
  - je sais appliquer l'algorithme de De Casteljau pour évaluer un point d'une courbe de Bézier ;
  - Je sais subdiviser une courbe de Bézier ;
  - je sais calculer la floraison d'un polynôme ;
  - Je connais les 3 propriétés caractérisant les floraisons ;
  - A l'aide des floraisons, je sais expliquer l'algorithme de De Casteljau, pour l'évaluation et pour la subdivision.
  - je sais détecter qu'une courbe de Bézier a une mauvaise paramétrisation.

# CTD 1

## Espaces de points

### Motivations

Avant d'étudier les courbes, nous devons regarder en détail leurs fondements : les points. Une courbe, ou une surface, est définie par un ensemble de points. Ces points seront paramétrés par un paramètre dans le cas des courbes ou deux paramètres dans le cas des surfaces.

Nous avons besoin de connaître la structure algébrique qui correspond aux ensembles de points. Que donne une structure algébrique ?

- Elle permet de définir/spécifier les éléments de l'ensemble,
- elle définit les opérations qui permettent de manipuler ces éléments.

### Compétences

Après avoir vu les espaces de points, vous devez répondre "oui" à ces questions :

- je sais définir un espace affine et je connais les éléments d'un espace affine, et je sais quelles opérations s'appliquent sur ces éléments ;
- je sais définir une base de l'espace affine, et je sais aussi définir un repère de cet espace ;
- je sais comparer les espaces vectoriels et les espaces affines ;
- je connais les limitations de la modélisation affine, et je sais que pour certaines transformations, il faut se tourner vers des espaces plus généraux : les espaces projectifs.
- Trouver les coordonnées barycentriques d'un point d'un espace affine dans une base affine de cet espace ;
- Etant données des coordonnées barycentriques d'un point sur une famille affine, dessiner le point dans l'espace affine ;
- Caractériser une transformation affine, algébriquement et géométriquement.

### 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** On peut représenter un point  $A$  ou un vecteur  $\overrightarrow{OA}$  de la même façon par leur coordonnées dans un repère (orthonormé) d'origine  $O$ , et donc considérer ces coordonnées comme un vecteur d'un espace vectoriel sur  $\mathbb{R}$ .

1. Quel est l'effet d'une translation sur le point  $A$  ? sur le vecteur  $\overrightarrow{OA}$  ?
2. Quel est l'effet d'une multiplication par un scalaire  $\lambda$  sur le point  $A$  ? sur le vecteur  $\overrightarrow{OA}$  ?

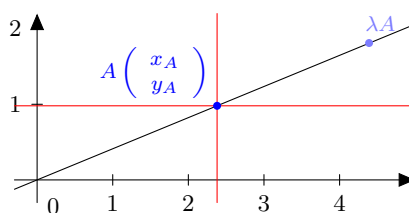
**Quelle structure algébrique pour les espaces de points ?** Pourquoi les espaces vectoriels ne suffisent pas ? Pourquoi est-il important de distinguer les points des vecteurs ? Si on prend un repère, les coordonnées sont définies dans ce repère. Et on pourrait se dire que le point  $A$  ou le vecteur  $\overrightarrow{OA}$  ont les mêmes coordonnées. Donc nous pourrions considérer que travailler sur  $A$  ou  $\overrightarrow{OA}$  est équivalent. Et pourtant,  $\overrightarrow{OA}$  et  $A$  sont différents.

Voici deux raisons pour lesquelles il faut distinguer les points des vecteurs. Ces raisons sont basées sur la définition même des espaces vectoriels :

**Définition :**  $E$  est appelé espace vectoriel si :  $\forall x, y \in E, \forall \lambda \in \mathbb{K}, \lambda x \in E, x + y \in E$ .

Tout d'abord, considérons une **translation**. L'effet de la translation est de bouger un point par un vecteur translation. Mais une translation sur un vecteur  $\vec{OA}$  n'a pas d'effet car un vecteur est la classe d'équivalence de bipoints équipotents.

Un autre exemple, la **multiplication d'un vecteur par un scalaire**  $\lambda$ , noté  $\lambda A$ , a un sens : c'est un vecteur de même direction et d'orientation que  $OA$  et une norme  $\lambda \|\vec{OA}\|$ . Mais multiplier un point par un scalaire n'a pas de sens. Supposons que nous prenons le point  $\lambda A$  le point  $A'$  tel que  $\vec{OA'} = \lambda \vec{OA}$ . Cette définition n'est pas correcte parce qu'elle dépend de l'origine  $O$  de la frame. Pour une origine différente  $O'$ , le résultat serait différent. Donc, les points doivent être différents des vecteurs.



$\left. \begin{matrix} \lambda A \\ A + B \end{matrix} \right\}$  pas de sens car  $\lambda A$  dépend de l'origine.

## 1 Espace affine

### 1.1 Combinaisons affines et espaces affines

Les points appartiennent à un espace affine. Quand on étudie l'algèbre linéaire (l'espace des vecteurs), les espaces affines sont souvent caractérisés par un point et un vecteur de l'espace. Pour un point donné  $P$  dans un espace affine  $\mathcal{A}$ , tous les autres points de  $\mathcal{A}$  s'écrivent de la manière suivante :  $P + v$  où  $v$  est un vecteur de l'espace vectoriel  $V$  associé à  $\mathcal{A}$ . Notons que nous considérons dans ce cours que des espaces de dimensions finies.

Ici, nous définirons plutôt un espace affine, indépendamment de l'espace vectoriel associé. Comme un espace vectoriel est constitué de vecteurs stables par combinaison linéaire, un espace affine est un espace constitué de points stables par combinaison affine.

**Définition :** Une **combinaison affine** est une combinaison linéaire particulière telle que la somme des coefficients est égale à 1. Si  $A_i, i = 0, \dots, m$  est un ensemble de points dans un espace affine  $\mathcal{A}$ , alors

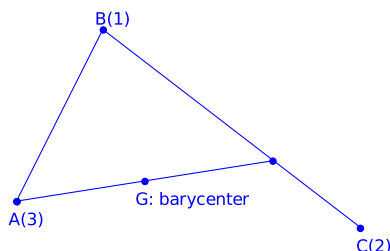
$$B = \sum_{i=0}^m \alpha_i A_i \text{ où } \sum_{i=0}^m \alpha_i = 1 \quad (1)$$

est aussi un point de l'espace affine  $\mathcal{A}$ .

Les combinaisons affines sont très liées à la notion de **barycentre**. Le point  $B$  défini par l'équation (1) est le barycentre des points  $A_i$  respectivement donnés par le poids  $\alpha_i$ . Ainsi, un espace affine peut être défini tel que, un barycentre de points dans cet espace appartient aussi à l'espace. Cette interprétation fournit une interprétation géométrique intuitive des combinaisons affines au moins pour des coefficients positifs.

Dans cette définition, le lien avec l'espace vectoriel associé peut être fait. Comme mentionné précédemment, l'espace vectoriel est l'espace de classes d'équivalence des points équipotents, c'est-à-dire les points séparés par le même vecteur (leur différence définit un même vecteur).

**Exemple** Un exemple classique d'un espace affine est l'espace euclidien de points 2D : tout barycentre de deux points du plan est un point du plan.



## 1.2 Arithmétique du point

Nous venons de définir les combinaisons affines et d'apprendre qu'une combinaison affine de points est un point. Nous pouvons définir un point dans un espace affine en utilisant le lien entre les points et les vecteurs. Le vecteur  $\overrightarrow{AB}$  est défini par la différence  $B - A$ . Donc on peut soustraire deux points et obtenir un vecteur. Plus généralement :

**Propriété** Toute combinaison linéaire de points telle que la somme des coefficients est nulle est un vecteur.

Considérons l'équation (1), on obtient

$$B = \sum_{i=0}^n \alpha_i A_i \text{ où } \sum_{i=0}^n \alpha_i = 1 \quad (2)$$

$$= \left( \sum_{i=0}^n \alpha_i \right) A_0 + \sum_{i=0}^n \alpha_i (A_i - A_0) \quad (3)$$

$$= A_0 + \sum_{i=0}^n \alpha_i \overrightarrow{A_0 A_i}. \quad (4)$$

Donc pour trouver le point  $B$ , il faut commencer par  $A_0$  et ajouter un à un les vecteurs  $\alpha_i \overrightarrow{A_0 A_i}$ .

**Pour résumer** Les combinaisons linéaires de points sont **valides** seulement si la somme des coefficients est égale à **1** (qui est une combinaison affine) ou si la somme des coefficients est **0**. Si la somme est 1, la combinaison est un **point**, si la somme est nulle, la combinaison est un **vecteur**.

## 1.3 Bases affines

**Définition :** Une **base** est un ensemble d'éléments tel que tout autre élément de l'espace peut être exprimé dans cette base. Cette expression doit être unique.

**Exemple** Dans le contexte des espaces vectoriels de dimension finie, une **base vectorielle** est un ensemble de vecteurs tel que tout élément de l'espace peut être défini par une combinaison linéaire de vecteurs dans cette base. Les coefficients de la combinaison linéaire sont appelés coordonnées et doivent être définis de manière unique.

Dans le cas des espaces vectoriels, nous avons la propriété que toutes les bases d'un espace ont le même nombre d'éléments ; ce nombre est appelé la dimension de l'espace.

**Base affine de points** De manière similaire, une **base d'un espace affine de dimension finie** est un ensemble de points, tel que tout point dans l'espace affine peut être défini comme une combinaison affine des points dans cette base. Les coefficients dans une combinaison affine sont les coordonnées de ce point et sont uniques. Intuitivement, ce point est le barycentre des points de la base avec des poids égaux aux coordonnées correspondantes.

Notons que cette propriété intuitive n'est pas une caractérisation dès lors que les poids peuvent être multipliés par une même constante sans changer le barycentre. L'unicité vient du fait qu'une combinaison affine force que la somme des poids égale à 1. Ces poids sont appelés les **coordonnées barycentriques** des points.

## Repère

Conformément à la définition d'un espace affine  $\mathcal{A}$  associé à l'espace vectoriel  $V$  :

$$\mathcal{A} = \{P' = P + v \mid P \text{ est un point donné, } v \in V\}.$$

Un repère de l'espace affine est plus naturellement défini par un point  $P$  appelé origine et une base de vecteurs de  $V$ . Le lien avec la définition précédente d'une base est que pour une base affine de points  $(A_i)_{0 \leq i \leq n}$  de  $\mathcal{A}$ , la base correspondante avec un point et  $n$  vecteurs est le point  $A_0$  et les vecteurs  $(\overrightarrow{A_0 A_i})_{1 \leq i \leq n}$ . C'est équivalent de dire que  $(A_i)_{0 \leq i \leq n}$  est une base affine de  $\mathcal{A}$  ou de dire que  $(\overrightarrow{A_0 A_i})_{1 \leq i \leq n}$  est une base vectorielle de  $V$ . À partir de cette équivalence, on peut déduire que le nombre de points dans toute base affine est  $n + 1$  où  $n$  est la dimension de l'espace vectoriel associé. Pour un point arbitraire  $B$  et avec l'équation (4), on obtient :

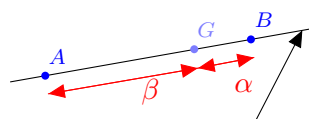
$$\begin{aligned} B &= \sum_{i=0}^n \alpha_i A_i \text{ avec } \sum \alpha_i = 1 \\ &= A_0 + \sum_{i=1}^n \alpha_i \overrightarrow{A_0 A_i} \end{aligned}$$

Donc les coordonnées affines de  $B$  dans la base affine  $(A_i)_{0 \leq i \leq n}$  sont  $(\alpha_0, \dots, \alpha_n)$  et les coordonnées cartésiennes de  $B$  dans la base cartésienne  $(A_0, \overrightarrow{A_0 A_1}, \dots, \overrightarrow{A_0 A_n})$  sont  $(\alpha_1, \dots, \alpha_n)$ . Dans ce qui suit, un vecteur  $\overrightarrow{AB}$  sera noté  $AB$ .

## Calculer des coordonnées barycentriques

**Exemple** Considérons deux points distincts  $A$  et  $B$  et un point  $G$  de la droite  $(AB)$ . Les deux points  $(A, B)$  forment une base de cette droite. On veut déterminer les coordonnées barycentriques  $(\alpha, \beta)$  de  $G$  à partir de  $A$  et  $B$  telles que :

$$G = \alpha A + \beta B \text{ where } \alpha + \beta = 1.$$



droite où se trouvent les barycentres

En fait, on peut vérifier avec l'arithmétique du point que les coordonnées sont définies par :

$$\begin{aligned}\overline{GB} &= \alpha \overline{AB} \\ \overline{AG} &= \beta \overline{AB}\end{aligned}$$

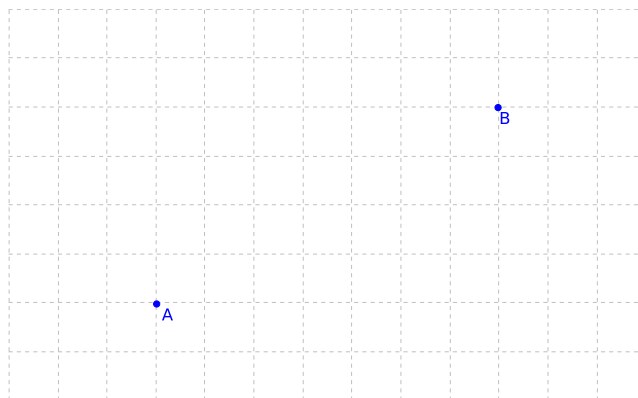
où  $\overline{AB}$  est la mesure algébrique de  $AB$ . Attention au fait que ces 2 relations ne sont pas si naturelles : si on considère que la longueur de  $AB$  est 1,  $\alpha$  est la distance signée de  $G$  à  $B$  et  $\beta$  est la distance signée de  $A$  à  $G$ . Il est important de penser en terme de barycentre, que le poids d'un point augmente à mesure que le barycentre se rapproche de ce point.

Si  $|AB| = 1$ ,  $\alpha = \frac{\overline{GB}}{\overline{AB}}$ ,  $\beta = \frac{\overline{AG}}{\overline{AB}}$ ,  $G = \alpha A + \beta B$  avec  $\alpha + \beta = 1$ .

### Exercice

Soient 2 points arbitraires distincts  $A$  et  $B$  définis dans une base affine. Sur la figure, tracer les points :

- $M1$  de coordonnées affines  $(1/2, 1/2)$ ,
- $M2$  de coordonnées affines  $(1/4, 3/4)$ ,
- $M3$  de coordonnées affines  $(-1, 1)$ .



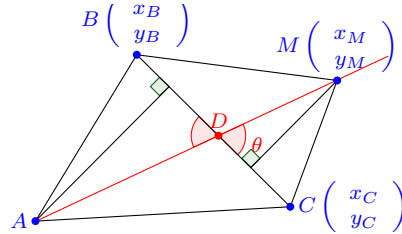
Maintenant, tracer 2 points arbitraires sur la ligne  $(AB)$ , un entre  $A$  et  $B$ , un à l'extérieur du segment  $[AB]$ . Trouver les coordonnées barycentriques de ces points dans la base  $A, B$ .



**Exemple avec 3 points** Dans le cas de 3 points  $A$ ,  $B$  et  $C$ , comme base affine du plan, un point  $M$  a pour coordonnées  $\alpha$ ,  $\beta$  et  $\gamma$  où :

- $\alpha$  est l'aire du triangle  $ABM$  sur l'aire du triangle  $ABC$ ,
- $\beta$  est l'aire du triangle  $BCM$  sur l'aire du triangle  $ABC$ ,
- $\gamma$  est l'aire du triangle  $CAM$  sur l'aire du triangle  $ABC$ ,

Trouver  $(\alpha, \beta, \gamma)$ .



**Résolution algébrique**  $\begin{cases} M = \alpha A + \beta B + \gamma C \\ \alpha + \beta + \gamma = 1 \end{cases} \rightarrow 3 \text{ équations à 3 inconnues.}$

**Résolution géométrique** Que valent  $D$  et  $M$  ?

$$D = \frac{\overline{DC}}{\overline{BC}}B + \frac{\overline{BD}}{\overline{BC}}C$$

$$M = -\frac{\overline{DM}}{\overline{AD}}A + \frac{\overline{AM}}{\overline{AD}}D = -\frac{\overline{DM}}{\overline{AD}}A + \frac{\overbrace{\overline{AM}}^{\alpha}}{\overline{AD}} \frac{\overline{DC}}{\overline{BC}}B + \frac{\overbrace{\overline{AM} \overline{BD}}^{\beta}}{\overline{AD}} \frac{\overline{BC}}{\overline{BC}}C$$

On peut exprimer la fraction  $\frac{\overline{DM}}{\overline{AD}}$  en fonction l'aire du triangle BMC

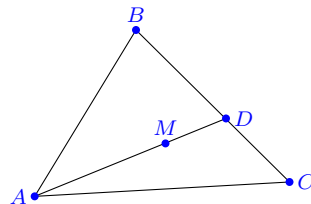
$$\frac{BMC}{ABC} = \frac{\overline{BC} \cdot \overline{DM} \sin(\theta)}{2} \frac{2}{\overline{BC} \cdot \overline{AD} \sin(\pi - \theta)} = \frac{\overline{DM}}{\overline{AD}}$$

donc :

$$\alpha = \frac{\overline{DM}}{\overline{AD}} = \frac{BMC}{ABC}$$

$$\gamma = \frac{ABM}{ABC}$$

$$\beta = \frac{AMC}{ABC}$$



Application numérique :

$$M = \frac{1}{2}A + \frac{1}{2}D$$

$$M = \frac{1}{2}A + \frac{1}{2} \underbrace{\left( \frac{2}{3}B + \frac{1}{3}C \right)}_D$$

## 1.4 Transformations affines

Comme les transformations linéaires sont des homomorphismes associés aux espaces vectoriels, les transformations affines sont des homomorphismes associés aux espaces affines.

### Définitions :

- Une transformation est **linéaire** si l'image d'une combinaison linéaire de vecteurs est la combinaison linéaire de l'image de ces vecteurs.
- Une transformation est **affine** si l'image d'une combinaison affine de points est la combinaison affine de l'image de ces points.

Intuitivement, cela signifie qu'une transformation affine conserve les barycentres :

$$f\left(\sum \alpha_i A_i\right) = \sum \alpha_i f(A_i) \text{ avec } \sum \alpha_i = 1. \quad (5)$$

Mais en fait, vous savez qu'une transformation affine s'écrit sous la forme :

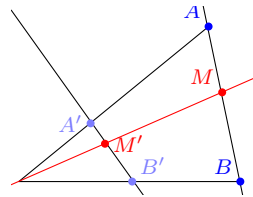
$$f(t) = at + b. \quad (6)$$

Les transformations données par l'équation (6) pour  $a$  et  $b$  arbitraires sont toutes des transformations affines de  $\mathbb{R}$  vers  $\mathbb{R}$ . On peut facilement vérifier la propriété caractéristique (5) pour ces fonctions.

Maintenant, considérons un espace affine de points dans le plan euclidien. Vous connaissez de nombreuses transformations dans cet espace :

- translations
- rotations,
- projection : orthogonale ou parallèle avec une direction arbitraire
- projection avec un centre de projection
- homothéties...

Sont-elles toutes affines ? Pour vérifier, nous avons besoin d'être sûr que les barycentres sont conservés. En fait, ce sont toutes transformations affines excepté la projection avec un centre de projection. On peut dessiner un exemple qui montre que les barycentres ne sont pas préservés. (Exemple : les selfies)



Pour définir une transformation linéaire, il est suffisant de connaître l'image d'une base. Les images de  $n$  vecteurs définissent la transformation linéaire. C'est aussi le cas de transformation affine, c'est suffisant de connaître les images de  $n + 1$  points d'une base affine.

### Une transformation affine est-elle linéaire ?

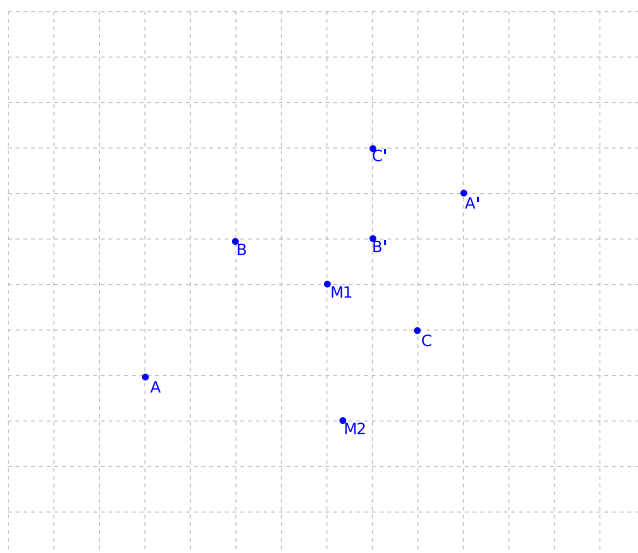
- Translation : affine mais non linéaire  
 $t(M) = M + \delta$   
 $t(\alpha M) = \alpha t(M)$   
 $t(\alpha M) = \alpha M + \delta \neq \alpha M + \alpha \delta = \alpha t(M)$
- Rotation, scaling : affine et linéaire

En fait : Linéaire  $\in$  Affine  $\in$  Projection.

affine	linéaire
points	vecteurs
espace affine	espace vectoriel
combinaisons affines	combinaisons linéaires
transformation affine	transformation linéaire

### Exercice

Les trois points  $A$ ,  $B$  et  $C$  définissent une base affine du plan, s'ils ne sont pas alignés. Une transformation affine planaire  $T$  est caractérisée par 3 points  $A'$ ,  $B'$  et  $C'$  étant les images respectives de  $A$ ,  $B$  et  $C$ . Dessiner les images  $M'_1$  et  $M'_2$  des points  $M_1$  et  $M_2$  dans le plan par la transformation  $T$ .



### 1.5 Espace de Grassmann ou coordonnées homogènes

Dans l'espace de Grassman, les points du plan sont décrits par 3 composantes :

$$\underbrace{\begin{pmatrix} x_A \\ y_A \\ 1 \end{pmatrix}}_{\text{coordonnée homogène}} = \begin{pmatrix} \lambda x_A \\ \lambda y_A \\ \lambda \end{pmatrix},$$

la troisième composante étant le poids. Cette notation permet de considérer des points à l'infini ou des vecteurs.

**Représentation matricielle :** on peut modéliser matriciellement les transformations par une matrice  $3 \times 4$  :  $3 \times 3$  pour la représentation linéaire, et le vecteur  $1 \times 3$  de translation dans la dernière colonne.

Dans une matrice  $4 \times 4$ , la dernière ligne peut être utilisée en géométrie projective pour exprimer une transformation perspective (encore plus générale, comme une photo).

# Paramétrisation

## Motivations

Maintenant que vous savez définir des points dans un espace affine, nous allons définir une courbe comme un ensemble de points. Ces courbes paramétrées peuvent être vues comme des fonctions à 1 paramètre, à valeurs dans un espace affine. Mais est-ce vraiment légitime d'associer une courbe à une telle fonction, appelée sa paramétrisation ?

Comme vous avez de grandes compétences en analyse fonctionnelle nous souhaitons faire le lien entre une courbe et une fonction la paramétrant. Nous allons voir dans quelles conditions cet amalgame (que nous ferons par la suite) est raisonnable.

## Compétences

Après avoir suivi ce bout de cours sur les paramétrisations, vous devez répondre "oui" à ces questions :

- je sais définir une courbe paramétrique ;
- je sais montrer qu'une paramétrisation est régulière, ou non ;
- je connais la paramétrisation idéale.

Je dois aussi savoir :

- trouver plusieurs paramétrisation d'une même courbe ;
- déterminer si une paramétrisation est bonne, ou non.

## 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 chaun vient avec une trace écrite de ses réflexions.

**La question** Soit la fonction  $f$  définie par :

$$\begin{aligned} f : \mathbb{R} &\rightarrow \text{plan affine} \\ t &\mapsto (t^2, -t^4) \text{ si } t \leq 0 \\ &\quad (t^2, t^2) \text{ sinon} \end{aligned}$$

Montrer que  $f$  est  $C^1$  (continue et de dérivée continue) partout (en particulier en 0).

Dessinez la courbe représentative de cette fonction paramétrique. Que remarquez vous quant à sa continuité ?

## 2 Représentations paramétriques – généralités

### 2.1 Définitions

Dans la suite, nous appellerons **courbe paramétrique** la représentation géométrique d'une fonction, définie sur un intervalle réel  $I$ , dont les valeurs appartiennent à un espace affine de dimension 2 ou 3.

Une **surface paramétrique** sera définie de manière similaire mais sur un intervalle de  $\mathbb{R}^2$ .

Les courbes et les surfaces peuvent être aussi définies de manière implicite comme étant les courbes/surfaces de niveaux d'une fonction potentiel de  $\mathbb{R}^2$  ou  $\mathbb{R}^3$  sur  $\mathbb{R}$ .

## 2.2 Différentes représentations

Il existe différentes façons de définir une courbe. La plus commune consiste à définir une courbe soit implicitement ou explicitement.

### Autre exemple introductif : la droite.

Prenons l'exemple d'une droite, et pensons aux différentes façons de la définir. Une droite est définie avec deux points  $A$  et  $B$ . Une façon de définir la droite  $(AB)$  est :

$$(AB) = \{M | \overrightarrow{AM} // \overrightarrow{AB}\} = \{M | \overrightarrow{AM} \cdot \vec{n} = 0\}$$

où  $\vec{n} = -(y_B - y_A, x_B - x_A)$  est un vecteur normal à la droite. Ce qui s'appelle une équation **implicite** de la droite : un point appartient à la droite si ses coordonnées vérifient l'équation.

Une autre façon de définir la droite est de considérer une équation paramétrique. La droite  $(AB)$  est définie par :

$$(AB) = \{M | M = A + t\overrightarrow{AB}\}$$

A chaque paramètre  $t$  dans  $\mathbb{R}$  correspond un point de la droite ; c'est une représentation **paramétrique** de la droite  $(AB)$ .

Les représentations implicite et paramétrique sont différentes et conviennent pour différentes opérations : par exemple, la représentation paramétrique est bien adaptée pour définir des points sur une surface. Ce n'est pas le cas pour la représentation implicite : trouver un point sur la courbe/surface n'est pas facile, mais étant donné un point, vérifier si ce point appartient à la courbe/surface est très facile. Ainsi, les modèles implicites sont pour la détection de collision. Dans un monde idéal, on aimerait avoir les deux représentations du même objet, mais ce n'est pas facile de passer de la représentation implicite à la paramétrique et réciproquement.

Il faut savoir qu'actuellement, 90 % des modèles CAO utilisent les courbes paramétriques et elles sont utilisées dans 50 à 60 % des jeux vidéo. Dans la suite, nous nous intéresserons principalement à la représentation paramétrique. Commençons par les courbes.

## 3 Courbes paramétriques

On utilise le terme de **courbe** pour une fonction paramétrisant une courbe, ce qui est incorrect car une courbe peut être paramétrisée par différentes fonctions. Par exemple :

$$(AB) = \{M | M = A + t\overrightarrow{AB}\} = \{M | M = A + t^3\overrightarrow{AB}\} = \{M | M = A + t\sin(t)\overrightarrow{AB}\}$$

décrivent la même droite. Par contre, la paramétrisation change la façon de décrire cette droite.

Dans la suite, on appellera **courbe paramétrique** la fonction paramétrique correspondant à la courbe même si c'est abusif. La courbe est interprétée comme une trace d'un parcours.

**Définition :** Les **courbes paramétriques** sont définies par une fonction réelle à valeur dans un espace affine  $\mathcal{A}$  :

$$\begin{aligned} f : \mathbb{R} &\rightarrow \mathcal{A} \\ t &\mapsto f(t) \end{aligned}$$

telle que  $f(t)$  est un point. Si on choisit un repère, avec un point (l'origine) et des vecteurs, alors on peut exprimer ces points par ses coordonnées cartésiennes. Parler de repères orthonormaux suppose que l'espace vectoriel a un produit scalaire. C'est le cas, en particulier, pour le plan euclidien. Alors, chaque coordonnée est une fonction de paramètre  $t$  et un point 2D s'écrit :

$$f(t) = (x(t), y(t)).$$

On a observé qu'une courbe peut avoir différentes paramétrisation. Si on imagine que la paramétrisation est une trajectoire, comme le chemin d'une mouche par exemple alors représenter la même courbe signifie que la trajectoire est la même mais la vitesse (dérivée première) à laquelle l'insecte vole peut être différente et dans ce cas, la paramétrisation est différente.

**La paramétrisation est-elle importante ?** Oui, une propriété à satisfaire est d'avoir une paramétrisation régulière.

### Définition :

Une paramétrisation  $F$  d'une courbe est dite **régulière** si, pour tout  $t$ ,  $F'(t) \neq 0$ .

$\Rightarrow$  La paramétrisation non régulière c'est mal!! :)

### Exercice

Regardons l'exemple suivant. Soit la fonction  $f$  définie par :

$$\begin{aligned} f : \mathbb{R} &\rightarrow \text{plan affine} \\ t &\mapsto \begin{cases} (t^2, -t^4) & \text{si } t \leq 0 \\ (t^2, t^2) & \text{sinon} \end{cases} \end{aligned}$$

Dessiner la courbe correspondante et montrer que cette fonction est  $C^1$  partout mais la paramétrisation est non régulière.

Ce que nous remarquons c'est que la paramétrisation peut être  $C^1$  partout mais la courbe peut ne pas avoir une tangente continue. En d'autres termes, la continuité de la fonction n'assure pas la continuité de la courbe. La représentation paramétrique n'assure pas la propriété géométrique. Parce que la représentation paramétrique n'est pas régulière. Avec une paramétrisation non régulière, on ne peut pas avoir le problème inverse c'est-à-dire une courbe lisse et une paramétrisation non  $C^1$ .

**Exercice :** Trouver un exemple de courbe avec une tangente continue mais qui n'a pas une paramétrisation  $C^1$ .

Avoir une paramétrisation régulière assure que la continuité de la fonction correspond à la continuité de la courbe. Ainsi, il est possible d'étudier la continuité de la courbe en regardant la continuité de la paramétrisation.

Plus généralement, les propriétés d'une courbe indépendante de la paramétrisation sont appelées **propriétés géométriques**. La **continuité géométrique de la tangente à une courbe** est appelée  $G^1$ . Être  $G^1$  est équivalent à avoir une paramétrisation uniforme c'est-à-dire  $C^1$ .

## 3.1 Paramétrisation longueur d'arc et tangente

C'est la paramétrisation idéale : la paramétrisation uniforme. Elle correspond à une vitesse constante (ou dérivée première constante), plus précisément, un vecteur vitesse égal à 1. Cette paramétrisation est appelée **paramétrisation en abscisse curviligne**.

Cette paramétrisation est un bon outil théorique s'il permet d'exprimer facilement ces quantités. En pratique, la paramétrisation longueur d'arc est difficile à obtenir.

La **tangente d'une courbe** est donnée par la dérivée  $f'(t)$ . La dérivée  $f'(t)$  est la limite de  $\frac{f(t+\epsilon) - f(t)}{\epsilon}$  quand  $\epsilon$  tends vers 0. Une différence de point est un vecteur, donc la tangente est un vecteur.

**Définition :** La **longueur d'une courbe** entre deux points  $f(t_0)$  et  $f(t)$  sur la courbe est donnée par :

$$s(t) = \int_{t_0}^t \|f'(u)\| du.$$

Si la paramétrisation est régulière,  $\|f'(u)\| = ds/dt$  est non nulle donc on peut faire un changement de variables entre  $t$  et  $s$ . Si on paramétrise la courbe par  $s$  à la place de  $t$  (changement de variable), on peut paramétriser la longueur d'arc, noté  $f(s)$ . Le vecteur  $\frac{df}{ds}$  est le vecteur tangent unitaire (car  $d(s^{-1})/du (= dt/ds) = 1/\|f'(u)\|$ ).

### 3.2 Plan osculateur et courbure

Nous considérons maintenant la dérivée seconde de la courbe, notée  $f''(t)$ . Elle mesure la variation de la courbe par rapport à la direction tangentielle.

Considérons  $f(s)$ , la paramétrisation longueur d'arc. Comme  $\|df/ds\| = (df/ds).(df/ds) = 1$ , alors  $(df/ds).(d^2f/ds^2) = 0$  (par dérivation).

**Définition :** on appelle **courbure**  $\kappa(t) = \|d^2f/ds^2\|$ . Avec une paramétrisation arbitraire,

$$\kappa(t) = \frac{\|f'(t) \times f''(t)\|}{\|f'(t)\|^3}.$$

Le **rayon de la courbure** est donnée par  $R = \frac{1}{\kappa}$ .

La dérivée première et seconde définissent un plan, appelé **plan osculateur**. Ce plan approche le mieux la courbe.

### 3.3 Torsion

**Définition :** la variation de la courbe sur le plan osculateur est mesurée par la **torsion** :

$$\tau(t) = \frac{\det(f(t), f'(t), f''(t))}{\|f'(t) \times f''(t)\|}.$$

De termes de paramètre  $s$ , la torsion peut exprimée plus facilement par :

$$\tau(s) = \frac{\det(f(s), df/ds(s), d^2f/ds^2)}{\kappa(s)^2}.$$

La courbure  $\kappa$  et la torsion  $\tau$  sont indépendants de la paramétrisation.

On veut être capable de décrire des courbes lisses intuitivement. La première façon est de donner quelques points et définir une courbe passant par ces points c'est-à-dire interpoler les points. La deuxième façon est de donner une idée de la forme de la courbe en donnant un polygone de contrôle c'est-à-dire une courbe linéaire par morceaux décrivant la forme que l'on souhaite générer. On parle alors respectivement d'interpolation et d'approximation.



## 4 CTD 2 : Interpolation de Lagrange

### Motivations

Nous allons dans cette partie revoir les courbes interpolantes de Mr Berger, avec deux nouvelles approches :

- On considère des courbes paramétriques,
- On voit une construction des fonctions interpolantes (Algorithme de Neville).

### Compétences

Suite au cours avec Mr Berger, vous devez répondre "oui" à ces questions :

- Etant donnés des points  $(x_i, y_i)$   $i = 0 \dots n$  je sais définir explicitement une courbe  $y = f(x)$  polynomiale de degré  $n$  qui passe par ces points.
- Je sais définir les polynômes de Lagrange relatifs à une famille de points, et montrer qu'il forment une base de l'espace des polynômes de degré inférieur ou égal à  $n$ .

Suite à ce CTD, vous devez "oui" à ces questions :

- je sais définir une courbe paramétrique polynomiale de degré  $n$  passant pas des points  $(x_i, y_i)$   $i = 0 \dots n$  ;
- Je sais que la courbe ainsi définie dépend des temps de passage choisi  $t_i$  pour chaque point à interpoler ;
- Je sais construire le polynome interpolant, itérativement à partir de l'algorithme de Neville.

Bonus :

- je sais définir une surface paramétrique polynomiale de bi-degré  $n$  passant pas des points  $(x_{i,j}, y_{i,j})$   $i, j = 0 \dots n$  explicitement ;
- Je sais que la surface ainsi définie dépend de temps d'interpolation (je dois savoir combien, et comment les définir et les utiliser pour définir les fonctions de base) et j'ai expérimenté l'influence de ces temps de passage sur la courbe interpolante ;
- Je sais aussi construire cette courbe interpolante itérativement en utilisant l'algorithme de Neville.

### 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 chaun vient avec une trace écrite de ses réflexions.

**La question : interpolation paramétrique** Soient  $A$  et  $B$  deux points. Définir une équation paramétrique  $l_{AB}(t)$  de la droite  $(AB)$  telle que  $l_{AB}(0) = A$  et  $l_{AB}(1) = B$ .

Plus généralement, peut-on paramétrer la droite  $(AB)$  telle que pour deux temps arbitraires  $t_A$  et  $t_B$  :

$$l_{AB}(t_A) = A, l_{AB}(t_B) = B?$$

Pouvez vous proposer une solution pour trouver une courbe interpolant maintenant trois points  $P_0, P_1, P_2$  aux temps  $t_0, t_1, t_2$  ?

## 4.1 Interpolation de Lagrange - version fonctionnelle

Soient  $(n + 1)$  points dans  $\mathbb{R}^2 : (x_i, y_i), i = 0..n$ . Comment déterminer un polynôme réel passant exactement par tous ces points ?

**Théorème de Lagrange :** Il existe un unique polynôme  $P_n$ , de degré inférieur ou égal à  $n$  tel que :

$$P_n(x_i) = y_i$$

$P_n$  est appelé **polynôme d'interpolation** et est défini par :

$$P_n(x) = \sum_{i=0}^n L_i(x) y_i$$

où  $L_i(x), i = 0..n$  représentent les **polynômes de Lagrange** de degré  $n$  défini par :

$$L_i(x) = \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j}.$$

Les polynômes de Lagrange vérifient, en particulier, que  $L_i(x_j) = \delta_{ij}$  où  $\delta$  représente le symbole de Kronecker.

## 4.2 Interpolation de Lagrange - version paramétrique

Un petit retour paramétrique sur l'interpolation...

On considère maintenant  $n + 1$  points  $P_i = (x_i, y_i), i = 0..n$  de  $\mathbb{R}^2$  et les paramètres associés  $(t_i)_{i=0..n}$ . Comment déterminer une fonction polynomiale  $F$  passant exactement par tous ces points  $P_i$  aux instants  $t_i$  respectifs c'est-à-dire telle que  $F(t_i) = (x(t_i), y(t_i))$  pour tout  $i = 0..n$  ?

**Théorème :** La fonction paramétrique  $F$  polynomiale de degré minimum telle que  $F(t_i) = (x_i, y_i)$  est définie par :

$$F(t) = \sum_{i=0}^n L_i(t) P_i$$

où

$$L_i(t) = \prod_{j=0, j \neq i}^n \frac{t - t_j}{t_i - t_j}.$$

sont les polynômes de Lagrange paramétriques.

**Remarques :**

- 1) Les polynômes de Lagrange forment une base des polynômes de degré  $n$ , et en ce sens, les points  $P_i$  sont les coefficients de  $F$  dans la base de Lagrange (base de l'espace de polynômes de degré inférieur ou égal à  $n$  –on démontre en exercice que les polynômes de Lagrange forment une base) ;
- 2) On peut aussi interpréter  $F(t)$  une combinaison affine qui définit le point  $F(t)$  comme barycentre des points  $P_i$  auquel on associe le poids  $L_i(t)$  (car la somme des polynôme de Lagrange fait toujours 1 –on démontre cela en exercice).

## 4.3 Choix des $t_i$ et paramétrisation

Il faut noter que les fonctions  $L_i(t)$  sont complètement dépendantes des valeurs choisies pour  $t_i$ . Déjà, l'ordre des  $(t_i)_i$  détermine l'ordre dans lesquels les points sont visités, mais aussi, le temps relatif pour parcourir le trajet entre deux points interpolés change la courbe.

Plus précisément, une famille  $(t'_i)_i$  donnera la même courbe que la famille  $(t_i)_i$  si, pour tout  $i$  :

$$t'_i = at_i + b.$$

On pourra choisir judicieusement les paramètres  $t_i$  suivant la longueur des segments :

$$t_{i+1} = t_i + ||P_i P_{i+1}||$$

ou encore, suivant la racine de la longueur des segments

$$t_{i+1} = t_i + \sqrt{||P_i P_{i+1}||}.$$

Le choix optimal est donné par une paramétrisation utilisant les abscisses de Chebychev (ou Tchebycheff) :

$$t_i = \cos \left( \frac{(2i+1)\pi}{2n+2} \right), \quad i = 0 \dots n.$$

(vu par Philippe Berger).

#### 4.4 Algorithme de Neville

Nous proposons ici un calcul direct et constructif pour calculer une fonction interpolante sans passer par les polynômes de Lagrange, mais en exploitant les combinaisons affines. Considérons d'abord, le cas de deux et trois points.

##### 4.4.1 Interpolation de deux points

Soient  $A$  et  $B$  deux points. La droite  $(AB)$  peut être paramétrisée par

$$l_{AB}(t) = (1-t)A + tB$$

Avec cette paramétrisation de la droite  $(AB)$ ,  $l_{AB}(0) = A$  et  $l_{AB}(1) = B$ .

Plus généralement, peut-on paramétrer la droite  $(AB)$  telle que pour deux temps arbitraires  $t_A$  et  $t_B$  :

$$l_{AB}(t_A) = A, l_{AB}(t_B) = B?$$

Oui, voilà la paramétrisation :

$$l_{AB}(t) = \frac{t_B - t}{t_B - t_A} A + \frac{t - t_A}{t_B - t_A} B.$$

$$\begin{array}{ccc} A & & B \\ & \searrow & \nearrow \\ \frac{t_B - t}{t_B - t_A} & & \frac{t - t_A}{t_B - t_A} \\ & \nearrow & \searrow \\ & l_{AB}(t) & \end{array}$$

Notez que  $l_{AB}$  est bien définie :  $l_{AB}(t)$  est une combinaison affine de  $A$  et  $B$  et la somme des coefficients vaut 1. Le point  $l_{AB}(t)$  ainsi défini, est donc le barycentre de  $A$  et  $B$  avec les poids respectifs  $\frac{t_B - t}{t_B - t_A}$  et  $\frac{t - t_A}{t_B - t_A}$  (attention à l'ordre... pas 'naturel' a priori).

#### 4.4.2 Interpolation aux trois points

Interpolons maintenant trois points  $P_0, P_1, P_2$  aux temps  $t_0, t_1, t_2$ . On ne peut pas interpoler trois points non alignés avec une droite, d'où la nécessité d'un plus grand degré de liberté (degré 2).

On procède comme précédemment avec  $P_0, P_1$  puis  $P_1, P_2$ . On définit la droite passant par  $P_0$  en  $t_0$ , et  $P_1$  en  $t_1$  :

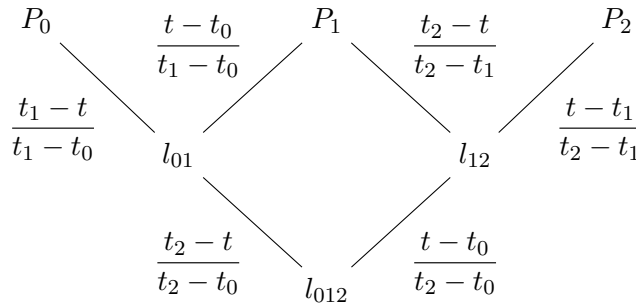
$$l_{01}(t) = \frac{t_1 - t}{t_1 - t_0} P_0 + \frac{t - t_0}{t_1 - t_0} P_1$$

puis la droite passant par  $P_1$  en  $t_1$ , et  $P_2$  en  $t_2$  :

$$l_{12}(t) = \frac{t_2 - t}{t_2 - t_1} P_1 + \frac{t - t_1}{t_2 - t_1} P_2.$$

Maintenant, on définit une fonction  $l_{012}$  en combinant de la même façon  $l_{01}$  et  $l_{12}$  en les associant à  $t_0$  et  $t_2$  :

$$l_{012}(t) = \frac{t_2 - t}{t_2 - t_0} l_{01}(t) + \frac{t - t_0}{t_2 - t_0} l_{12}(t)$$



A faire comme exercice : On a :

$$l_{012}(t_i) = P_i, \quad i = 0 \dots 2.$$

On note que la fonction  $l_{012}$  est de degré 2, car  $l_{01}$  et  $l_{12}$  sont de degré 1 et les coefficients sont eux aussi de degré 1.

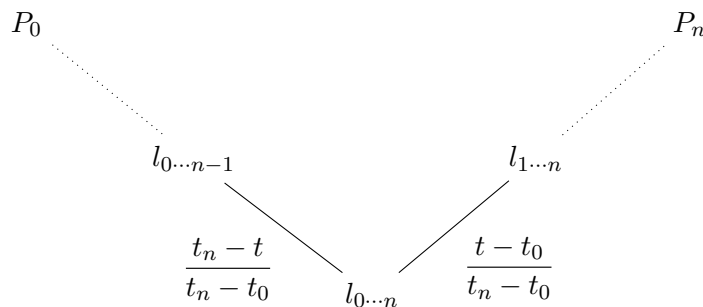
#### 4.4.3 Interpolation à $n + 1$ points

Interpolons  $P_0 \dots P_n$  aux temps  $t_0 \dots t_n$  respectivement.

Par récurrence, on utilise la même idée que pour trois points.

$$l_{0\dots n} = \frac{t_n - t}{t_n - t_0} l_{0\dots n-1}(t) + \frac{t - t_0}{t_n - t_0} l_{1\dots n}(t)$$

C'est sur ce principe que se fonde l'**algorithme de Neville** (algorithme de programmation dynamique).



On constate que l'algorithme de Neville permet de déterminer la fonction paramétrique  $F$  polynomiale de degré minimum telle que :  $F(t_i) = (x_i, y_i) \in \mathbb{R}^2$ . Par unicité, cette fonction est bien celle définie précédemment dans la base de Lagrange.

**Problème :** Si le degré  $n$  augmente, les oscillations de la fonctions interpolante entre les points  $P_i$  augmentent. De ce fait, une interpolation de haut degré de donnera pas une courbe suffisamment lisse. On préférera donc souvent des courbes approximantes... (comme au prochain CTD).

**Remarque :** Si on souhaite interpoler non seulement les valeurs d'une fonction, mais aussi contraindre ses dérivées, alors il faut regarder l'interpolation avec les **polynômes de Hermite** (cf cours de Ph. Berger).

#### 4.5 Surface interpolante : produit tensoriel

On considère maintenant une surface paramétrique  $S$  :

$$S: \begin{array}{ccc} \mathbb{R}^2 & \rightarrow & A \\ (s, t) & \mapsto & S(s, t) \end{array}$$

où  $A$  est un espace affine de dimension 3. On définit pour cela une grille de points de contrôles 3D de taille  $(n+1) \times (m+1)$  et deux familles de paramètres  $(s_i)_{i=0..n}$  et  $(t_j)_{j=0..m}$  tel que :  $S(s_i, t_j) = P_{ij}$ ,  $\forall i \in \{0..n\}$ ,  $j \in \{0..m\}$ .

La **surface interpolante de Lagrange en produit tensoriel** est alors le polynôme de degré inférieur ou égal à  $n$  en  $s$  et à  $m$  en  $t$  défini, pour tout  $s, t \in \mathbb{R}^2$ , par :

$$S(s, t) = \sum_{i=0}^n \sum_{j=0}^m L_i(s) L_j(t) P_{ij}$$

où  $L_i(s)$  et  $L_j(t)$  représentent les polynômes de Lagrange de degré respectivement  $n$  et  $m$ .

On peut considérer cette surface comme une courbe de courbes, i.e. :

$$S(s, t) = \sum_{i=0}^n L_i(s) P_i(t)$$

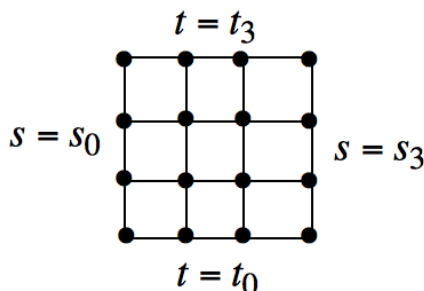
avec  $n+1$  points  $P_i(t)$  pris sur  $n+1$  courbes (à  $t$  fixé, pour  $i = 0 \dots n$ ) :

$$P_i(t) = \sum_{j=0}^m P_{ij} L_j(t).$$

**Exemple :** prenons un exemple avec une grille  $4 \times 4$  points de contrôles décrites par la figure suivante. On cherche une surface paramétrique lisse  $S(s, t)$  telle que :

$$S(s_i, t_j) = P_{ij} \in \mathbb{R}^3, \forall i, j \in \{0, \dots, 3\}$$

On trace le domaine des paramètres  $(s_i, t_j)$  associé à la grille des points de contrôle  $P_{ij}$



$P_{03}$	$P_{13}$	$P_{23}$	$P_{33}$
$P_{02}$	$P_{12}$	$P_{22}$	$P_{32}$
$P_{01}$	$P_{11}$	$P_{21}$	$P_{31}$
$P_{00}$	$P_{10}$	$P_{20}$	$P_{30}$

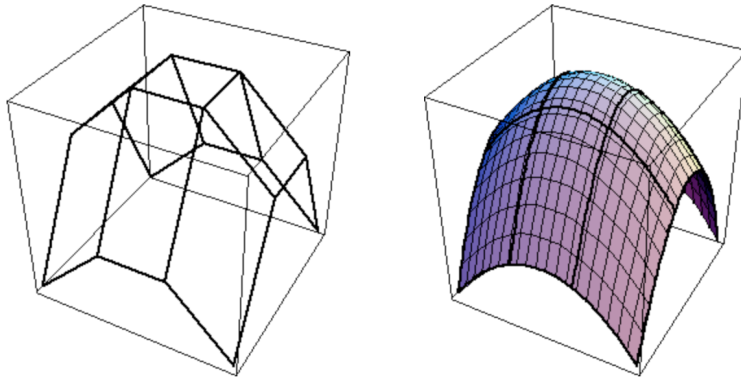
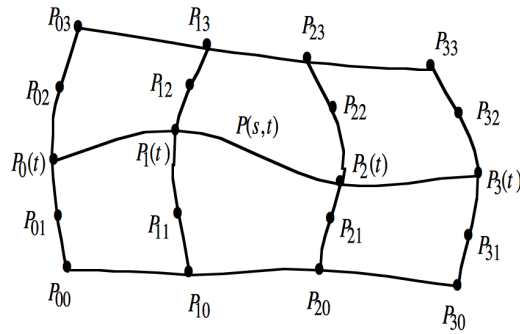


FIGURE 1 – Exemple de surface interpolante

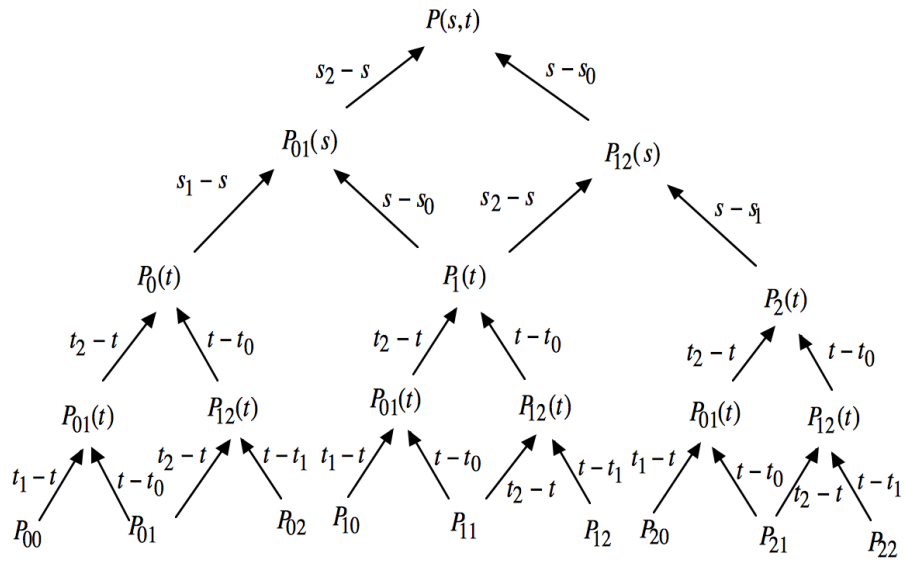


Surface paramétrique : domaine, grille de points de contrôle 3D et paramétrisation

La surface interpolante de Lagrange en produit tensoriel pour cet exemple s'écrit alors :

$$S(s, t) = \sum_{i=0}^3 \sum_{j=0}^3 P_{ij} L_i(s) L_j(t), \quad \forall s, t \in [0, 1]$$

**Version algorithmique :** L'algorithme de Neville pour les surfaces interpolantes revient à appliquer l'algorithme de Neville sur les lignes puis sur les colonnes. Voici un exemple sur une grille  $3 \times 3$  de points de contrôles 3D :



## CTD2.2 + CTD3 : Courbes de Bézier

### 5 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

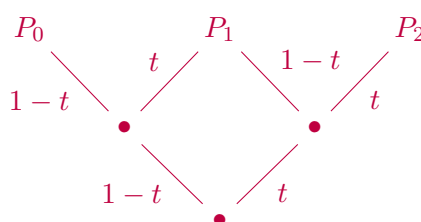
Après avoir vu les deux prochains CTDs, vous devez répondre "oui" à ces questions :

- 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 chaun 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 ?

#### 5.1 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 developpement de Taylor en 0 ! Cette information est donc très localisée.

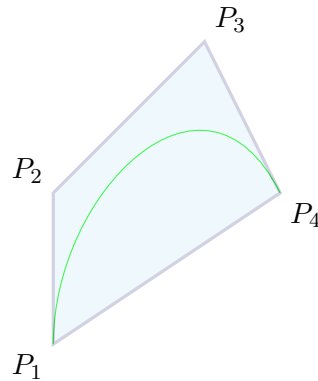
La base de Bernstein est préférée aux bases de monômes pour son interprétation géométrique plus globale : les courbes de Bézier ont une expression de courbe polynômiale où le coefficient correspond à des points de contrôle qui caractérisent la forme de la courbe.



**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.

## 5.2 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**  $(P_0...P_n)$  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  $[0, 1]$ .

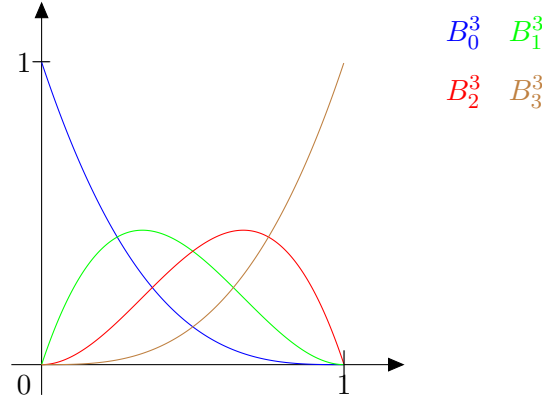
**Définition :** on appelle **Polynômes de Bernstein** les polynômes, pour tout  $t \in [0, 1]$ , définis par :

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

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

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

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



**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 :

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

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...(vois section 5.5).

### 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 approcher une courbe de Bézier par  $L$ , l'erreur maximale est bornée par la distance des points de contrôle à la ligne.

### Formule de Wang

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

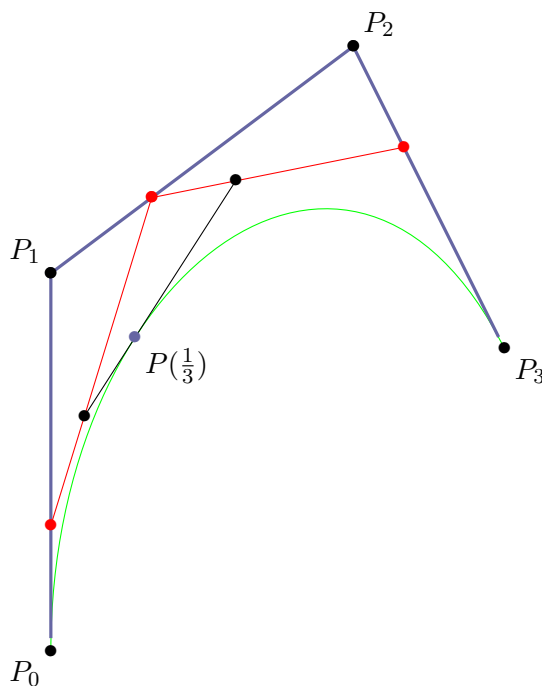
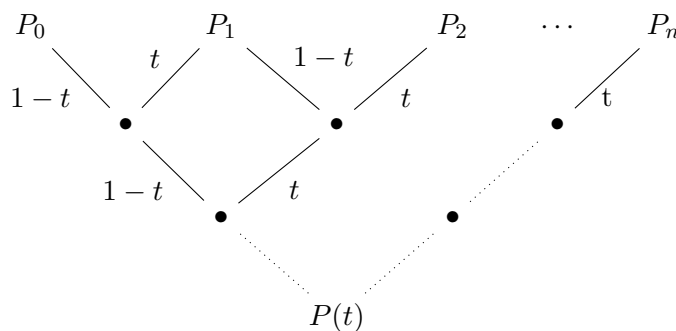
où  $\|\Delta_2 P\| = |P_{k+2} - 2P_{k+1} + P_k|$  La différence entre les points de contrôle correspond aux points de contrôle de la dérivée seconde de  $P$ .

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.

### 5.3 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.

## 5.4 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.

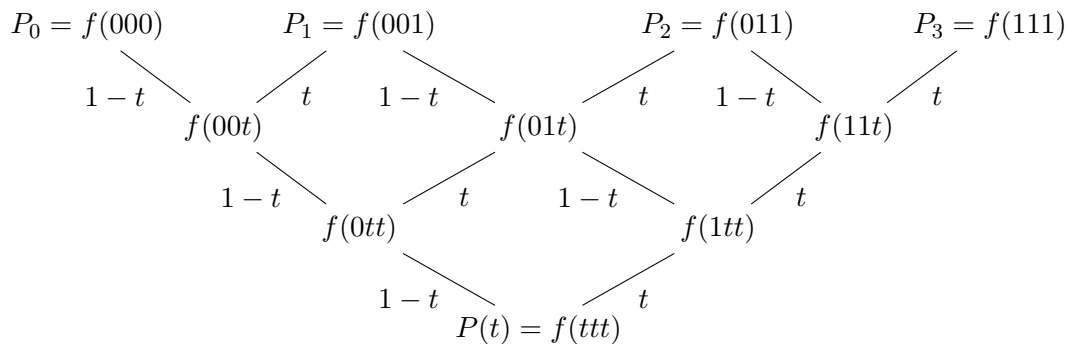
- Tous ces espaces sont des espaces vectoriels de dimension  $n + 1$ .
- Soit  $f \in \mathcal{F}_n$  alors pour  $x_1 \dots x_n \in \mathbb{R}$  on a  $f((1 - \lambda)u + \lambda v, x_2 \dots x_n) = (1 - \lambda)f(u, x_2 \dots x_n) + \lambda f(v, x_2 \dots x_n)$  qui est affine sur la première variable.
- $f(x_{\sigma(1)} \dots x_{\sigma(n)}) = f(x_1 \dots x_n)$  pour une permutation  $\sigma$  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  $b$  :  $P_i = f(\underbrace{a \dots a}_{n-i \text{ fois}}, \underbrace{b \dots b}_i)$ .

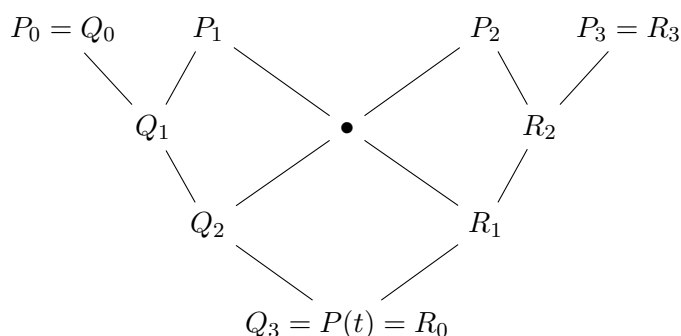
La notation de floraison sera aussi particulièrement utile pour les B-splines et leurs points de contrôle.

## 5.5 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.

## 5.6 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 2).

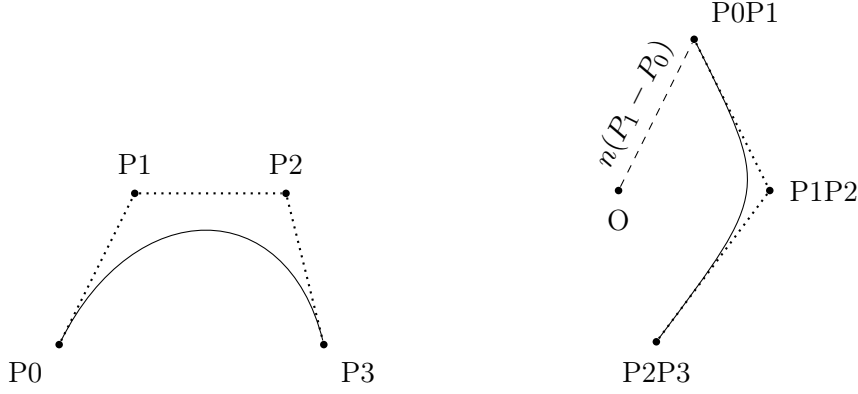


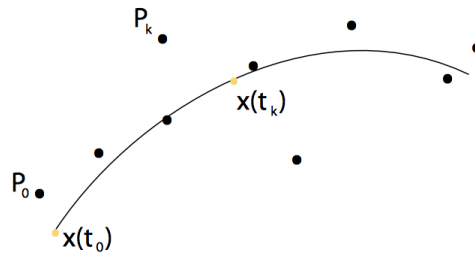
FIGURE 2 – Une courbe de Bézier (à gauche) et son hodographe (à droite).

**Propriétés :**

- la floraison de  $P'$  est  $t \mapsto n(\underbrace{p(t, \dots, t, 1)}_{n-1 \text{ fois}}) - \underbrace{p(t, \dots, t, 0)}_{n-1 \text{ fois}})$  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.

## 5.7 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 :

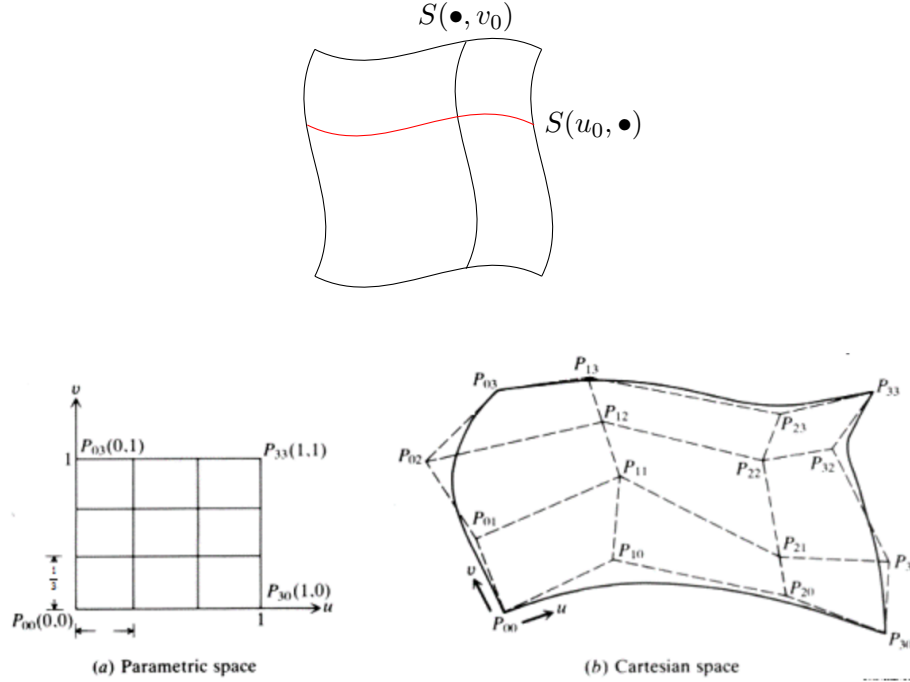
$$\min_Q \sum_{k=1}^n \|P_k - x(t_k)\|_2^2$$

## 5.8 Surfaces de Bézier

Surface paramétrique de Bézier :

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

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  $P_i$ .

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

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

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

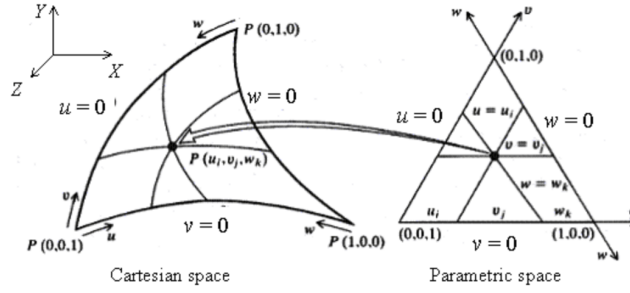
Calculer les surfaces de Bézier par produit tensoriel fonctionne bien. On travaille d'abord sur les lignes puis sur les colonnes : on travaille en **variables séparées**.

**Ecriture matricielle des surfaces de Bézier** :

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



— **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.

## CTD4 : Courbes polynomiales par morceaux

### Motivations

Nous avons vu que les interpolants souffraient d'oscillations, et que les polynômes perdaient les effets de forme lorsque l'on considère des degrés trop élevés. Nous allons donc proposer un modèle qui marie le meilleur des 2 mondes : il permet de gagner des degrés de liberté, conserver les bonnes propriétés des courbes de Bézier tout en permettant de préserver des effets de forme. De plus, il est important de noter que les fonctions de Bernstein sont à support global, c'est à dire qu'elles ne sont nulles que sur un ensemble discret (au plus  $n$  points puisque c'est des polynômes non nuls de degré  $\leq n$ ). Comme se sont les poids associés à un point de contrôle, cela veut dire que ce point de contrôle a une influence sur toute la courbe (son poids n'est jamais nul). De ce fait, si on bouge ce point de contrôle, il est nécessaire de recalculer toute la courbe. Nous allons voir qu'un autre avantage des splines, est qu'un point de contrôle n'affecte qu'une partie limitée de la courbe, car le support des fonctions de base est borné.

Ces modèles de B-splines, sont très utilisés et sont devenus des standards de l'industrie et logiciels de CAO.

### Compétences

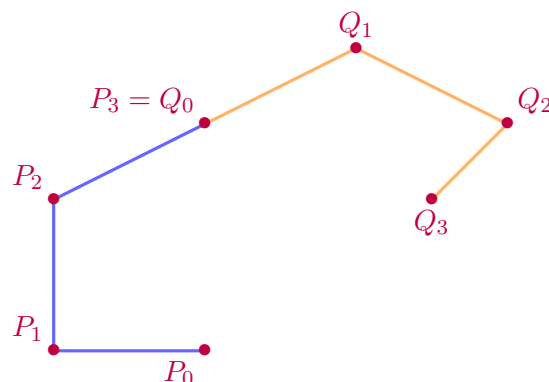
Après avoir vu le prochain CTD, vous devez répondre "oui" à ces questions :

- Je sais définir une spline polynomiale par son degré et son vecteur de noeuds ;
- Je sais exprimer les points de contrôle, évaluer un point sur une courbe spline, ou subdiviser, en utilisant les foraisons ;
- Je sais simplifier et coder l'algorithme de subdivision pour les courbes splines uniformes ;
- Je sais coder la subdivision pour les splines uniformes de surfaces en produit tensoriel.

### 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 chaun vient avec une trace écrite de ses réflexions.

**La question : Joignons 2 courbes de Bézier** Si on met bout à bout deux courbes de Bézier de degré 3, quelles sont les conditions sur les points de contrôle pour que le raccord des deux courbes soit  $C^0$  ?  $C^1$  ?  $C^2$  ?  $C^3$  ?  $C^4$  ?



## 6 Splines

### Limitations des courbes de Bézier / polynômes

— Degré élevé, peu d'effet de formes

— Contrôle global : un point de contrôle change  $\Rightarrow$  toute la courbe change.  $\Rightarrow$  Contrôle local et plus de formes

Contrôle local donc  $F(t) = \sum_k \underbrace{P_k}_{\text{points de contrôle}} \underbrace{f_k(t)}_{\text{fonctions de base}}$ . Influence des points de contrôle

$\Rightarrow$  support compact de  $f_k(t)$ .

Solution : polynômes par morceaux.

## 6.1 Introduction

Correction de l'activité préalable...

$$P(1) = P_3 \quad P'(1) = 3(P_3 - P_2) \quad P''(1) = 3 \times 2(P_1 - 2P_2 + P_3)$$

$$Q(0) = Q_0, \quad Q'(0) = 3(Q_1 - Q_0), \quad Q''(0) = 3 \times 2(Q_0 - 2Q_1 + Q_2),$$

$$P'''(1) = Q'''(0) \Rightarrow \text{même polynôme.}$$

Supposons qu'on considère une base de polynômes  $f_k(t)$  par morceaux telle que :  $F(t) = \sum_k P_k f_k(t)$ .

## 6.2 Définition d'une spline

Comme pour les courbes de Bézier qui sont définies dans la base de Bernstein, il existe une base "géométriquement intuitive" pour les polynômes par morceaux, appelée base de B-splines ; là aussi, les coefficients dans cette base sont des points de contrôle de la courbe. Par contre, nous ne définirons pas explicitement ces fonctions de base, car leur expression est compliquée, et inutile en pratique. Nous utiliserons d'une part, l'algorithme de de Boor pour évaluer un point sur la courbe (idem que de Casteljau-évaluation) en se basant sur les propriétés de la floraison. D'autre part, nous verrons qu'il existe un algorithme de subdivision simple, et très facile à implémenter.

**Définition :** Une **spline polynômiale**  $S$  est définie par :

— Son degré  $n$

— Ses noeuds rangés dans un vecteur de noeuds  $V$ .  $V = \{\dots t_k, t_{k+1}, t_{k+2} \dots\}, t_k \leq t_{k+1}$ .

Un noeud est un paramètre de discontinuité (paramètres  $t_k$  où on autorise des discontinuités).

**Propriété :** Si  $S$  est un polynôme sur  $[t_k, t_{k+1}]$  et si  $t_k$  apparaît  $i$  fois dans  $N$  alors  $S(\cdot)$  est  $\mathcal{C}^{n-i}$  en  $t_k$ .

**Exemple :** degré 3.  $V = \{\underbrace{000}_{C^0} \underbrace{1}_{C^2} \underbrace{2}_{C^2} \underbrace{33}_{C^1} \underbrace{444}_{C^0}\}$

**Théorème :** Etant donné un degré et un vecteur de noeuds existe une unique fonction  $N_k^n(t)$  de support minimum  $[t_k, t_{k+n+1}]$  définie à une constante près pour un degré  $n$  donné.

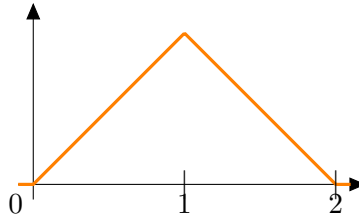
— On peut normer les  $(N_i^n)_i$  tels que  $\forall t, \sum_i N_i^n(t) = 1$ .

— Les  $(N_i^n)_i$  forment une base de l'espace des splines.

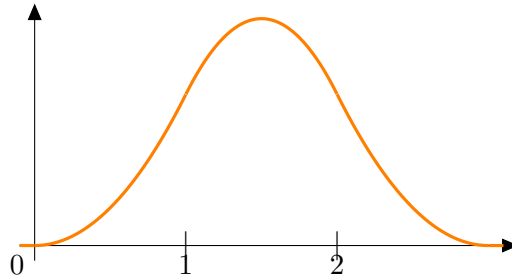
— On définit les points de contrôle de la spline  $P_i$  comme les coefficients dans la base  $(N_i^n)_i$ , c'est à dire,

$$S(t) = \sum_i P_i N_i(t)$$

**Exemple de polynôme de degré 1 :** continuité  $C^0$  en  $i \in \mathbb{Z}$  et support compact minimum :  $[0, 2]$ .



**Exemple polynôme de degré 2 :** continuité  $C^1$  en  $i \in \mathbb{Z}$  et support compact minimum :  $[0, 3]$ .



### 6.3 Points de contrôle et floraison

**Définition :** Les points de contrôle (coefficients dans la base des B-splines) sont définis par la floraison  $s$  de  $S$  en  $n$  valeurs consécutives de  $V$  :

$$P_k = s(t_{k+1} \dots t_{k+n})$$

**Exemple :** Considérons le degré 2,  $V = \mathbb{Z}$ . Points de contrôle  $P_i = s(i+1, i+2)$ .  $(P_i)_i$  donnés. Que valent les  $S(i)$  ?

1.  $S(1)$  ?  $S(1) = s(1, 1)$  (propriété de la diagonale)  
et on a :  $s(1, 1) = 1/2 s(0, 1) + 1/2 s(1, 2)$   
ou encore sous forme algorithmique :

$$\begin{array}{ccc} s(0, 1) & & s(1, 2) \\ & \searrow \frac{1}{2} & \swarrow \frac{1}{2} \\ & s(1, 1) & \end{array}$$

2.  $S(3/2) = s(3/2, 3/2)$   
on calcule d'abord  $s(1, 3/2) = \frac{1}{4} s(0, 1) + \frac{3}{4} s(1, 2)$   
puis :  $s(3/2, 2) = \frac{3}{4} s(1, 2) + \frac{1}{4} s(2, 3)$   
et enfin :  $s(3/2, 3/2) = 1/2 s(1, 3/2) + 1/2 s(3/2, 2)$ .  
ou encore sous forme algorithmique :

$$\begin{array}{ccccccc} s(0, 1) & & & s(1, 2) & & & s(2, 3) \\ & \searrow \frac{1}{4} & & \swarrow \frac{3}{4} & & \searrow \frac{3}{4} & \swarrow \frac{1}{4} \\ & s(1, 3/2) & & & & s(3/2, 2) & \\ & & \searrow \frac{1}{2} & & \swarrow \frac{1}{2} & & \\ & & s(3/2, 3/2) & & & & \end{array}$$

## 6.4 B-splines uniformes

**Définition :** Les splines sont appelées **uniformes** quand le vecteur de noeuds  $N$  est  $\mathbb{Z}$ . On a alors  $N_k^n(t) = N_0^n(t - k)$ , c'est à dire, toutes les fonctions sont translatées les unes des autres. Dans ce cas, subdiviser revient à passer d'un vecteur de noeuds  $= \mathbb{Z}$  à un vecteur de noeuds  $= \mathbb{Z}/2$ . [un cadre plus généra de subdivision a été développé récemment dans le cadre des NURBS, Cashman 2009].

On peut utiliser la floraison pour trouver les coefficients de subdivision. Regardons comment trouver les nouveaux points de contrôle qui sont associés à  $\mathbb{Z}/2$  que l'on note  $(\tilde{P}_i^n)_{i \in \mathbb{Z}/2}$ , à partir des « vieux » points de contrôle (ceux de  $\mathbb{Z}$ ), notés  $(P_i^n)_{i \in \mathbb{Z}}$ . Cela correspond à une étape de subdivision.

**Degré 2 :** On dispose d'un ensemble de points de contrôle.  $P_i^2 = s(i+1, i+2)$  et on cherche un ensemble de nouveaux points de contrôle  $\tilde{P}_i^2 = s(i+1/2, i+1)$  ou  $\tilde{P}_{i+1/2}^2 = s(i+1, i+3/2)$ . On a (première étape avec les floraisons) :

$$\begin{aligned} P_0^2 &= s(1/2, 1) \\ &= \frac{3}{4} \underbrace{s(0, 1)}_{P_{-1}} + \frac{1}{4} \underbrace{s(1, 2)}_{P_0} . \\ P_{1/2}^2 &= s(1, 3/2) \\ &= \frac{1}{4} \underbrace{s(0, 1)}_{P_{-1}} + \frac{3}{4} \underbrace{s(1, 2)}_{P_0} . \end{aligned}$$

Comme la spline est uniforme, les mêmes calculs s'appliquent partout (quel que soit  $i$ ).

**Degré 2 :** en disposant des  $(P_i^2)_i$ .

$$\begin{cases} \tilde{P}_i^2 = s(i+1/2, i+1) = \frac{3}{4}P_{i-1}^2 + \frac{1}{4}P_i^2 \\ \tilde{P}_{i+1/2}^2 = s(i+1, i+3/2) = \frac{1}{4}P_{i-1}^2 + \frac{3}{4}P_i^2 \end{cases}$$

*Faire une figure, avec un polygone qui contient des points  $(P_i^2)_i$  et en déduire (dessiner) les nouveaux points de contrôle.*

On remarque deux choses : les nouveaux points sont 2 fois plus 'denses' que les anciens. Le nouveau polygone de contrôle est plus 'lisse' que le premier.

**Attention** Comme le polynôme, la floraison est définie par morceaux, et donc il faut partir des 'bons' points de contrôle pour calculer les nouveaux. Si vous trouvez des coordonnées barycentriques négative, c'est que vous n'avez pas considéré les bons points de départ.

Pour **degré 1** : à partir des  $(P_i^1)_i$ .  $P_i = s(i+1)$ .

$$\begin{cases} \tilde{P}_i^1 = \frac{1}{2}P_{i-1}^1 + \frac{1}{2}P_i^1 \\ \tilde{P}_{i+1/2}^1 = P_i^1 \end{cases} \quad \text{Ce cas correspond aux fonction linéaires par morceaux, et donc le}$$

polygone de contrôle ne change pas après la subdivision (mais le nombre de points est doublé).

**Degré 3 :** On dispose des  $(P_i^3)_i$ .  $P_i^3 = s(i+1, i+2, i+3)$  et on cherche l'ensemble de nouveaux points de contrôle  $(\tilde{P}_i^n)_{i \in \mathbb{Z}/2}$ .

*A faire en exercice.*

**Degré 3 :** on dispose de  $(P_i^3)_i$ .

$$\begin{cases} \tilde{P}_{i+1/2}^3 = \dots \\ \tilde{P}_i^3 = \dots \end{cases}$$

Faire une figure, avec un polygone qui contient des points  $(P_i^2)_i$  et en déduire (dessiner) les nouveaux points de contrôle.

Il y a une façon générale de définir des nouveaux points de contrôle à partir des anciens. On peut remarquer que les points de degré 3 sont en fait le milieu de points de degré 2 et de même entre les points de degré 2 et ceux de degré 1.

Dans le cas général  $\tilde{P}_i^{n+1} = \frac{1}{2}\tilde{P}_{i+1/2}^n + \frac{1}{2}\tilde{P}_i^n$ .

L'algorithme des courbes B-splines de cette méthode de subdivision est très simple pour des B-splines uniformes : elle contient 2 étapes.

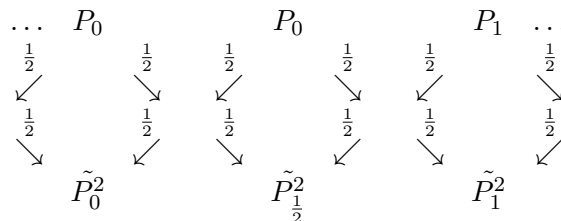
1. Dupliquer les points de contrôle (subdivision au degré 0).
2. Faire  $n$  fois (pour un degré  $n$ ) : Prendre le milieu de deux points consécutifs dans le polynôme de de contrôle.

La deuxième étape est répétée autant de fois que le **degré**  $n$  de la spline.

Comme dans Bézier, la subdivision peut être itérée pour avoir une meilleure approximation de la courbe. Ces 2 étapes sont donc à répéter autant de fois que le nombre de subdivisions demandés.

Avec la notation des floraisons, on obtient :

1.  $(\dots, P_{-1}, P_0, P_1, \dots) \rightarrow (\dots, P_{-1}, P_{-1}, P_0, P_0, P_1, P_1, \dots)$
2. Par exemple pour le degré 2

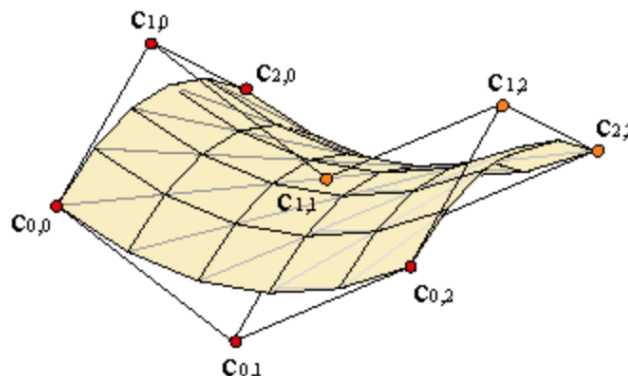


**Propriétés :** les splines ont des propriétés similaires aux courbes de Bézier.

- Elles suivent la forme du polynôme de contrôle.
- Elles restent à l'intérieur de l'enveloppe convexe des points de contrôle.
- Propriété de diminution de la variation.

spap2 : Least-squares spline approximation (matlab)

## 6.5 Surface B-spline : produit tensoriel



Comme pour les surfaces de Bézier, les surfaces B-spline en produit tensoriel est une extension des courbes B-splines vers deux directions que sont les paramètres  $u$  et  $v$ .

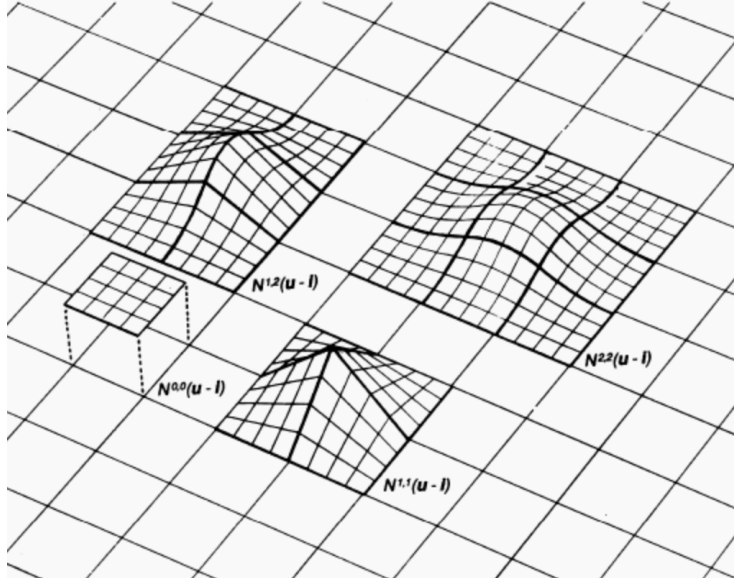
**Définition :** La surface B-spline par produit tensoriel est définie par :

$$S(u, v) = \sum_{i=1}^n \underbrace{\sum_{j=1}^m P_{ij} N^{jl}(u) N^{ik}(v)}_{P_i(u)}, \quad \forall u, v \in [0, 1]$$

La surface B-spline peut être considérée comme une courbe de courbes  $P_i(u)$  : on procède aussi en **variables séparées**.

**Ecriture matricielle des surfaces B-Splines :**

$$S(u, v) = [N^{0k}(u) \dots N^{nk}(u)] \begin{bmatrix} P_{00} \dots P_{0m} \\ \vdots \\ P_{n0} \dots P_{nm} \end{bmatrix} \begin{bmatrix} N^{0l}(v) \\ \vdots \\ N^{ml}(v) \end{bmatrix}$$



Exemples de surface B-Spline par produit tensoriel

$$N^{r,s}(u-i) = N^r(u-i)N^s(v-l)$$

## 7 Surfaces de subdivision

Les **surfaces de subdivision** sont basées fondamentalement sur un maillage de base auquel on fixe des règles de subdivision c'est-à-dire un schéma de subdivision.

A partir de ce schéma de subdivision, l'**évaluation récursive** revient à répéter deux étapes consécutives :

1. raffinement linéaire
2. lissage

L'**évaluation paramétrique** revient à :

1. construire une surfaces spline pour chaque polygone de base en fonction de son voisinage et du schéma de subdivision
2. évaluer la surface limite de subdivision en évaluant la surface spline (coordonnées barycentriques).

**Quelques définitions :**

- Sommet pair : sommet existant à la précédente étape de subdivision
- Sommet impair : sommet inséré à l'étape courante
- sommet régulier : valence 6 pour pour les maillages triangulaires, 4 pour les maillages quadrangulaires
- Sommet extraordinaire : tous les autres sommets
- Schéma dual : subdivision de faces
- Schéma primal : subdivision des sommets.

Schéma	Type	Maillage Triangulaire	Maillage Quadrangulaire
Primal	Approximation	Loop	Catmull-Clark
Primal	Interpolation	Butterfly modifié	
Dual	Approximation	$\sqrt{3}$	Doo-Sabin

Classification des surfaces de subdivisions

### 7.1 Matrice de subdivision

Un schéma de subdivision peut être exprimé par une matrice  $M$  de poids  $\omega$  :

- $M$  est creuse pour les schémas usuels
- $M$  ne doit pas être utilisée pour l'implémentation

Ce schéma permet l'étude analytique de la surface limite et en particulier de ses propriétés de continuité (tangentes, courbures ...)

$$P^{i+1} = MP^i$$

$$\begin{pmatrix} \omega_{00} & \omega_{01} & \dots & 0 \\ \omega_{10} & \omega_{11} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} p_0^i \\ p_1^i \\ \vdots \\ p_n^i \end{pmatrix} = \begin{pmatrix} p_0^{i+1} \\ p_1^{i+1} \\ \vdots \\ p_n^{i+1} \end{pmatrix}$$

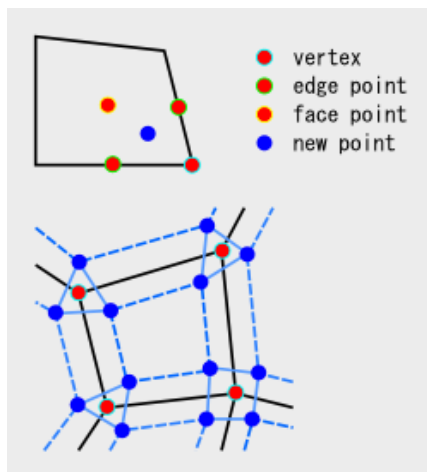
En fait, un schéma est exprimé par 2 matrices, une de raffinement pour insérer les nouveaux sommets et une de convergence pour déplacer les sommets existants (la plus intéressante à étudier).



## 7.2 Doo Sabin

L'algorithme de Doo-Sabin est une subdivision de spline par produit tensoriel de degré 2. Si vous faites des splines de degré 2 en produit tensoriel, vous obtenez l'algorithme de Doo-Sabin pour le cas régulier (tous les sommets sont réguliers).

- Point sur la face : moyenne entre les sommets du polygone
- point sur une arête : moyenne de l'arête
- nouveau point : moyenne des 4 points (point de la face, 2 points de l'arête et le sommet)



## 7.3 Schéma Catmull-Clark

Le schéma de Catmull-Clark est un schéma approximant pour maillages quadrangulaires avec un raffinement primal. La continuité est  $C^2$  partout sauf aux sommets extraordinaires ( $C^1$ ). La façon classique d'exprimer l'algorithme de Catmull-Clark est la suivante :

$$f_j^{k+1} = \frac{1}{|F_j|} \sum_{f_i \in F_j} f_i^k$$

$$e_j^{k+1} = \frac{v^k + e_j^k + f_j^{k+1} + f_{j-1}^{k+1}}{4}$$

$$v^{k+1} = \frac{n-2}{n} v^k + \frac{1}{n^2} \sum_{j=0}^{n-1} e_j^k + \frac{1}{n^2} \sum_{j=0}^{n-1} f_j^{k+1}$$

- Pour chaque face, un nouveau sommet est créé en son centre.
- Pour chaque arête, un nouveau sommet est créé comme le barycentre entre le milieu de l'arête et les deux face points adjacents à l'arête
- déplacer les anciens sommets en utilisant le barycentre des  $k$  face point adjacents et le barycentre des  $k$  edge point adjacents pour déplacer le somme vers le sommet satisfaisant la formule suivante de  $v^{k+1}$ .

**Remarque :** Si vous faites des splines de degré 3 en produit tensoriel, vous obtenez l'algorithme de Catmull-Clark pour le cas régulier (tous les sommets sont réguliers).

**Forme factorisée** Comme pour le cas des courbes, on peut exprimer cet algorithme de subdivision plus simplement sous une forme factorisée (ça marche dans le cas régulier et aussi irrégulier) :

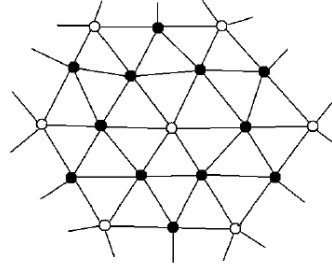
- on commence par effectuer une subdivision linéaire : chaque arête est coupée en 2, chaque face est coupée en 4.
- on calcule un barycentre par face (isobarycentre des 4 sommets)
- on remplace chaque sommet au barycentre des barycentres des faces qui le contiennent.

On remarque que les 2 dernières étapes peuvent être effectuées en un parcours des faces, lorsque l'on a calculé le barycentre  $baryc\_face$  de la face, on ajoute sa contribution à chacun des sommets  $v$  de la face :

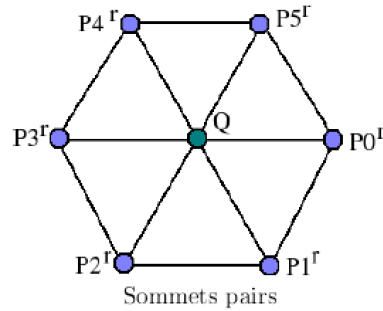
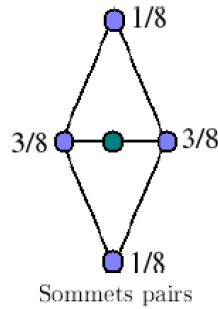
$$nouvelle\_position(v) += baryc\_face / valence(v).$$

## 7.4 Schéma Loop

Il s'agit d'un schéma approximant pour des maillages triangulaires avec un raffinement primal. La continuité est  $C^2$  par tout sauf aux sommets extraordinaires où la continuité est  $C^1$ .



Les masques de Loop :



Les coefficients des  $P_i^r$  sont  $\frac{\alpha_n}{n}$  et celui de  $Q^r$  est  $(1 - \alpha_n)$  avec  $n$  le nombre de voisins et

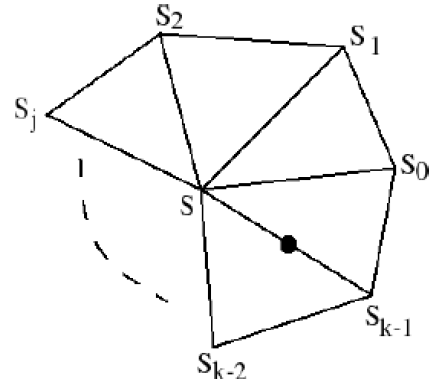
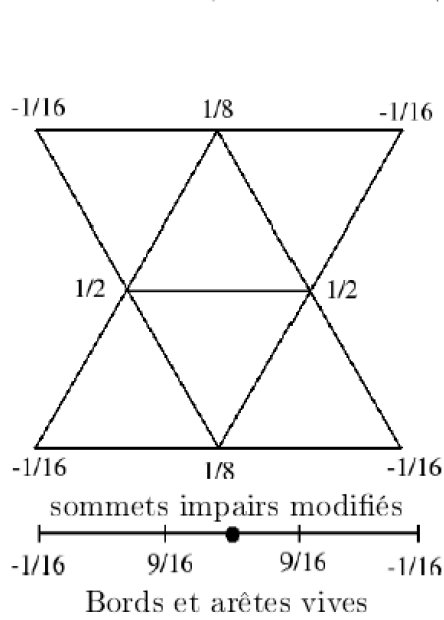
$$\alpha_n = \frac{1}{64} (40 - (3 + 2 \cos(\frac{2\pi}{n}))^2)$$

D'où

$$Q^{r+1} = (1 - \alpha_n)Q^r + \sum_{i=0}^n \frac{\alpha_n}{n} P_i^r$$

## 7.5 Butterfly modifié

Il s'agit d'un schéma interpolant pour maillage triangulaire qui est  $C^1$  partout avec cette version modifiée.



Sommets impairs extraordinaires

$$k = 3 \quad s_0 = \frac{5}{12} \quad s_{1,2} = \frac{-1}{12}$$

$$k = 4 \quad s_0 = \frac{3}{8} \quad s_2 = \frac{-1}{8} \quad s_{1,3} = 0$$

$$k > 5 \quad \frac{1}{4} \left( \frac{1}{4} + \cos \frac{2i\pi}{n} + \frac{1}{2} \cos \frac{4i\pi}{k} \right)$$

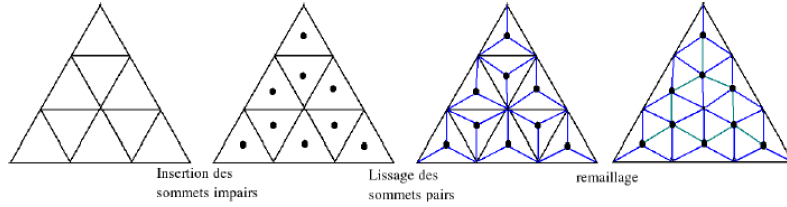
Note : comme ce schéma est interpolant, il y a uniquement des masques de sommets impairs

**Forme factorisée** Comme pour pour Catmull-Clark, on peut exprimer cet algorithme de subdivision plus simplement sous une forme factorisée (ça marche dans le cas régulier et aussi irrégulier) :

- on commence par effectuer une subdivision linéaire : chaque arête est coupée en 2, chaque face est coupée en 4.
- on calcule 3 barycentres par face (isobarycentre des 4 sommets)
- on replace chaque sommet au barycentre des barycentres des faces qui le contiennent.

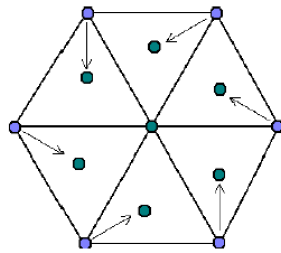
## 7.6 Racine de 3

C'est un schéma triadic (raffinement 1-3) pour maillage triangulaire (rotation de la topologie locale) avec une étape de raffinement plus fine que le schéma loop.



Insertion des sommets impair : on prend le barycentre du triangle :

$$q = \frac{1}{3}(p_i + p_j + p_k)$$



Relaxation des sommets pairs : on fait une interpolation entre la position d'origine et le barycentre du 1-voisinage :

$$S(p) = (1 - \alpha_n)p + \alpha_n \frac{1}{n} \sum_{i=0}^{n-1} p_j$$

avec

$$\alpha_n = \frac{4 - 2 \cos(\frac{2\pi}{n})}{9}$$