Cours Approximation Géométrique

Géraldine Morin, Sandrine Mouysset

19 mars 2020

1 CTD 2 : Interpolation de Lagrange

Motivations

Nous allons dans cette partie revoir les courbes interpolantes :
- On considère des courbes paramétriques,
- On voit une construction des fonctions interpolantes (Algorithme de Neville).

Compétences

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 chacun 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 ?

1.1 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 défni 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 δ représente le symbôle de Kronecker.

1.2 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 polynômiale 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 polynômiale 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).

1.3 Choix des t_i et paramétrisation

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

Plus précisement, 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 longeur des segments :

$$t_{i+1} = t_i + ||P_iP_{i+1}||$$

ou encore, suivant le racine de la longueur des segments

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

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).

1.4 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.

1.4.1 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 :

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

$$\begin{tikzcd} A \arrow[dr, dash, "\frac{t_B - t}{t_B - t_A}"'] & & B \arrow[dl, dash, "\frac{t - t_A}{t_B - t_A}"] \\ & l_{AB}(t) & \end{tikzcd}$$

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).

1.4.2 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 :

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

$$\begin{tikzcd}
P_0 \arrow[dr, dash, "\frac{t_1 - t}{t_1 - t_0}"'] & & P_1 \arrow[dl, dash, "\frac{t - t_0}{t_1 - t_0}"'] \arrow[dr, dash, "\frac{t_2 - t}{t_2 - t_1}"] & & P_2 \arrow[dl, dash, "\frac{t - t_1}{t_2 - t_1}"] \\
& l_{01} \arrow[dr, dash, "\frac{t_2 - t}{t_2 - t_0}"'] & & l_{12} \arrow[dl, dash, "\frac{t - t_0}{t_2 - t_0}"] & \\
& & l_{012} & &
\end{tikzcd}$$

A faire comme exercice : On a :

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

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.

1.4.3 Interpolation à $n + 1$ points

Interpolons $P_0 ... P_n$ aux temps $t_0 ... t_n$ 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).

$$\begin{tikzcd}
P_0 \arrow[dr, dotted, dash] & & & & P_n \arrow[dl, dotted, dash] \\
& l_{0 \cdots n-1} \arrow[dr, dash, "\frac{t_n - t}{t_n - t_0}"'] & & l_{1 \cdots n} \arrow[dl, dash, "\frac{t - t_0}{t_n - t_0}"] & \\
& & l_{0 \cdots n} & &
\end{tikzcd}$$

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 interpolnate entre les points P_i augmentent. De ce fait, une interpolantion de haut degré de donnera pas une courbe suffisamment lisse. On préfèrera 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).

2 Interpolation dans l'espace des quaternions

Les quaternions fournissent une notation mathématique alternative pour représenter l'orientation et la rotation d'objets en trois dimensions. Comparés aux angles d'Euler, ils sont plus simples à composer et évitent le problème du blocage de cardan. Comparés aux matrices de rotations, ils sont plus stables numériquement et se révèlent plus efficaces. Ils sont une généralisation des nombres complexes.

2.1 Construction d'un quaternion

On définit trois symboles i, j, k respectant les propriétés suivantes :

$$\begin{gathered} i^2 = -1 \\ j^2 = -1 \\ k^2 = -1 \end{gathered}$$

De plus la multiplication de deux de ces symboles se comporte comme le produit vectoriel de vecteurs unitaires orthogonaux :

$$\begin{gathered} i * j = -j * i = k \\ j * k = -k * j = i \\ k * i = -i * k = j \end{gathered}$$

Un quaternion est définit comme ci-dessous :

$$q = w + x * i + y * j + z * k$$

avec $(w, x, y, z) \in \mathcal{R}^4$

2.2 Propriétés

2.2.1 Conjugé d'un quaternion

On peut également définir le conjugé d'un qutaenrion en opposant toutes les composantes selon i, j et k :

$$\bar{q} = w - x * i - y * j - z * k$$

2.2.2 Norme d'un quaternion

On définit la norme d'un quaternion par :

$$||q|| = \sqrt{w^2 + x^2 + y^2 + z^2}$$

De plus :

$$||\bar{q}|| = ||q||$$
$$||q_1 * q_2|| = ||q_2 * q_1||$$

2.2.3 Associativité et commutativité

La multiplication de quaternion est **associative** :

$$(q_1 * q_2) * q_3 = q_1 * (q_2 * q_3)$$

La multiplication de quaternion **n'est absolument pas commutative** :

$$q_1 * q_2 \neq q_2 * q_1$$

2.3 Différentes représentations des quaternions

Il y a trois modes de représentation des quaternions :
- une combinaison linéaire de 1, i, j et k
- un vecteur de 4 composantes
- un couple composé d'un scalaire et d'un vecteur représentant respectivement la partie réelle et la partie imaginaire du quaternion.

On peut donc avoir :

$$q = w + x * i + y * j + z * k = [w, x, y, z] = (w, \vec{u})$$
$$\text{avec } \vec{u} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

2.4 Opérations possibles sur les quaternions

On note $q_1 = w_1 + x_1 * i + y_1 * j + z_1 * k$ et $q_2 = w_2 + x_2 * i + y_2 * j + z_2 * k$

2.4.1 Addition et soustraction

$$q_1 + q_2 = (w_1 + w_2) + (x_1 + x_2) * i + (y_1 + y_2) * j + (z_1 + z_2) * k$$
$$q_1 - q_2 = (w_1 - w_2) + (x_1 - x_2) * i + (y_1 - y_2) * j + (z_1 - z_2) * k$$

2.4.2 Multiplication

$q_1 * q_2 = (w_1 * w_2 - x_1 * x_2 - y_1 * y_2 - z_1 * z_2) * 1 + (w_1 * x_2 + x_1 * w_2 - y_1 * z_2 - z_1 * y_2) * i + (w_1 * y_2 - x_1 * z_2 + y_1 * w_2 + z_1 * x_2) * j + (w_1 * z_2 + x_1 * y_2 - y_1 * x_2 + z_1 * w_2) * k$

Remarque : On peut remarquer que si on utilise la représentation $(w_{ind}, \vec{u_{ind}})$, il est possible d'écrire le produit sous la forme :
$q_1 * q_2 = (w_1 * w_2 - (\vec{u_1}|\vec{u_2}), w_1 * \vec{u_2} + w_2 * \vec{u_1} + \vec{u_1} \wedge \vec{u_2})$

2.4.3 Multiplication par un scalaire

On a $\lambda * q = (\lambda * w, \lambda * x, \lambda * y, \lambda * z)$

2.4.4 Produit scalaire

On a $q_1 * q_2 = w_1 * w_2 + x_1 * x_2 + y_1 * y_2 + z_1 * z_2$

2.4.5 Inverse d'un quaternion

L'inverse d'un quaternion se définit par :

$$q * q^{-1} = q^{-1} * q = 1$$
$$q^{-1} = \bar{q}/||q||$$

2.5 Quaternions unitaires et rotations

2.5.1 Quaternions unitaires

On peut définir un quaternion unitaire en imposant la nomre de ce quaternion égale à 1. Il devient alors possible d'écrire :
$q = cos(\theta/2) + sin(\theta/2)\vec{u}$ avec la norme de u égale à 1.

2.5.2 Quaternion et rotation

Soit un quaternion unitaire $q = w + xi + yj + zk = w + \begin{pmatrix} x \\ y \\ z \end{pmatrix} = cos(\alpha/2) + sin(\alpha/2)\vec{u}$.

Soit $\vec{v}$ un vecteur de $\mathcal{R}^3$ sous forme de quaternion, c'est à dire avec une partie réelle nulle.

On peut montrer que $q\vec{v}q^{-1}$ permet d'effectuer une rotation d'angle α autour de l'axe dirigé par $\vec{u}$
De plus on peut composer les conjugaisons : $pq\vec{v}(pq)^{-1} = pq\vec{v}q^{-1}p^{-1} = p(q\vec{v}q^{-1})p^{-1}$

2.5.3 Exemple :

https ://drive.google.com/open ?id=1M9IG0tmzkYsUNszD41WmmV0s7H0UQQpU

2.6 Interpolation de quaternion

On peut interpoler des quaternions grâce à la méthode d'interpolation sphérique.
Si on considère $t \in [0, 1]$ et deux quaternions $q1, q2$ tel que l'angle entre $q1$ et $q2$ soit de Θ, on a

$$Slerp(t, q1, q2) = \frac{q1sin((1-t)\Theta)}{sin(\Theta)} + \frac{q2sin(t * \Theta)}{sin(\Theta)}$$

3 Surface interpolante : produit tensoriel

On considère maintenant une surface paramétrique S :

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

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 $i = 0 \ldots n$) :

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

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

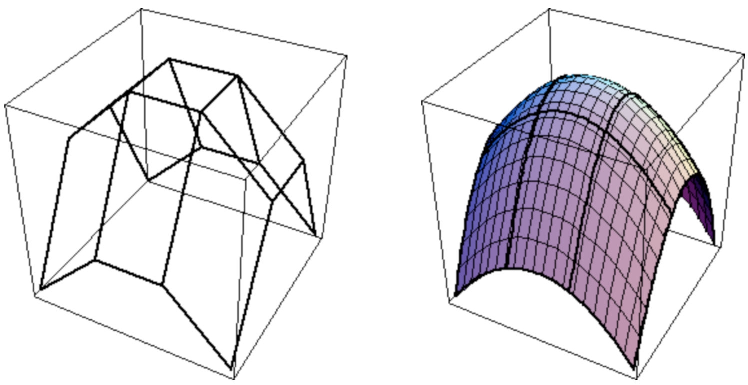
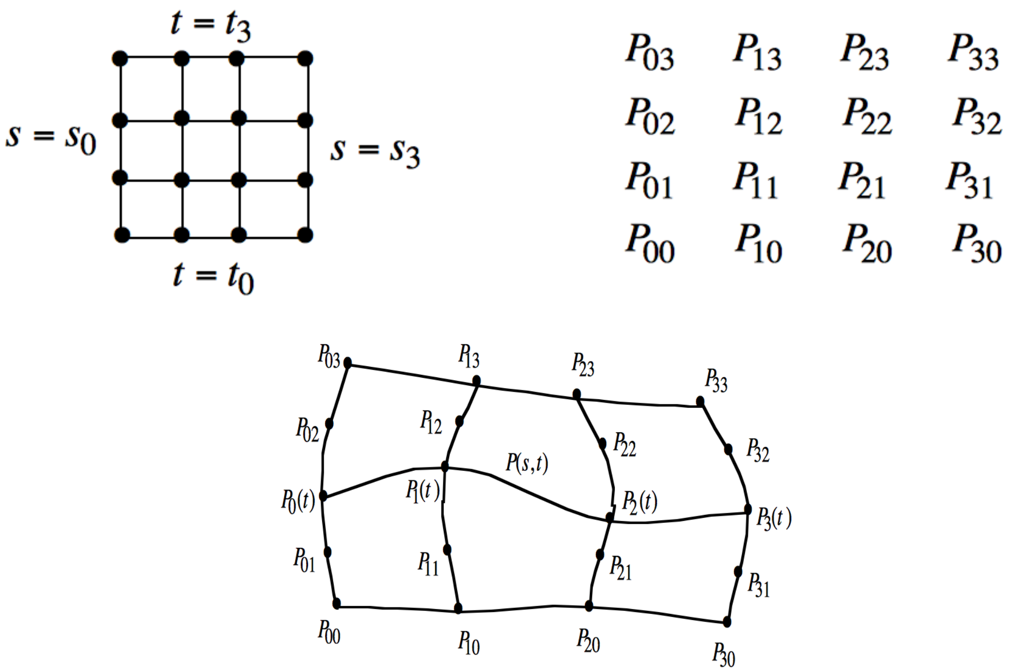


FIGURE 1 – Exemple de surface interpolante

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

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



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 :

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

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 3×3 de points de contrôles 3D :