TD 6 Courbes de Bézier et splines

I Courbes de Bézier

a. Renault et Citroën

Dans les années 60, les ingénieurs Pierre BÉZIER et Paul DE CASTELJAU travaillant respectivement chez Renault et Citroën, réfléchissent au moyen de définir de manière la plus concise possible la forme d'une carrosserie.
Le principe a été énoncé par BÉZIER mais l'algorithme de construction par son collègue de la marque aux chevrons qui n'a d'ailleurs été dévoilé que bien plus tard, la loi du secret industriel ayant primé sur le développement scientifique...
Aujourd'hui, les courbes de Bézier sont très utilisées en informatique.
Une Courbe de Bézier est une courbe paramétrique aux extrémités imposées avec des points de contrôle qui définissent les tangentes à cette courbe à des instants donnés.

b. Algorithme de Casteljau

i. Avec 3 points de contrôle

Soit t un paramètre de l'intervalle $[0,1]$ et P_1, P_2 et P_3 les trois points de contrôle.
On construit le point M_1 barycentre du système $\{(P_1, 1-t), (P_2, t)\}$ et M_2 celui du système $\{(P_2, 1-t), (P_3, t)\}$.
On construit ensuite le point M, barycentre du système $\{(M_1, 1-t), (M_2, t)\}$.
Exprimez M comme barycentre des trois points P_1, P_2 et P_3.
Faites la construction à la main avec $t = 1/3$ par exemple.

Commentez le programme suivant :

```
bezier1:=proc(p1,p2,p3)
local m,m1,m2,P,S,B,k,t;
P:=plot([p1,p2,p3],color=green);
B:=NULL;
for k from 0 to 1 by 0.01 do
   m1:=(1-k)*p1+k*p2;
   m2:=(1-k)*p2+k*p3;
   m:=(1-k)*m1+k*m2;
   B:=B,m;
od;
B:=plot([B],color=blue);
S:=seq(plots[display](B,P,plot([(1-t*0.01)*p1+t*0.01*p2
    ,(1-t*0.01)*p2+t*0.01*p3],color=red)),t=0..100);

plots[display](S,insequence=true,axes=none,scaling=
    constrained);
end:
```

puis le résultat de son lancement :

```
bezier1([0,0],[3,2],[5,-2]);
```

Il s'agit d'une animation ! Il faut donc cliquer sur l'image et lancer l'animation à l'aide de l'icône appropriée.

ii. Espace affine

Vous étudierez bientôt les espaces affines après avoir étudié les espaces vectoriels.

Avec un peu d'avance, nous allons juste y faire une incursion de manière très intuitive.
Un espace affine c'est en gros une sorte d'espace vectoriel « pointé ». Si un point A et un vecteur u sont donnés, $B = A + u$ est un autre élément de l'espace affine. Plus clairement, dans le contexte du plan vectoriel vu au lycée, on écrit :

$$B = A + \overrightarrow{AB}$$

Cela nous permet de faire des calculs sur les points. D'ailleurs MAPLE calcule lui aussi de cette manière.
Ainsi $M_1(t) = (1-t)P_1 + tP_2$ (ou si vous préférez $\overrightarrow{OM_1}(t) = (1-t)\overrightarrow{OP_1} + t\overrightarrow{OP_2}$) et $M_2(t) = (1-t)P_2 + tP_3$ puis

$$M(t) = (1-t)M_1(t) + tM_2(t) = (1-t)^2P_1 + 2t(1-t)P_2 + t^2P_3$$

On peut dériver $M(t)$ par rapport à t (ou $\overrightarrow{OM}(t)$ par rapport à t) comme en physique (vous verrez ça de manière rigoureuse en maths plus tard) ce qui nous donnera le vecteur vitesse instantanée (c'est-à-dire un vecteur directeur de la tangente).
Vérifiez que $M'(t) = 2(M_2(t) - M_1(t))$. Comment l'interpréter ?

iii. Avec 4 points de contrôle

Faites une étude similaire (« à la main ») avec 4 points de contrôle.
On pourra introduire les polynômes de Bernstein définis par :

$$B_j^n(t) = \binom{n}{j} t^j (1-t)^{n-j}$$

et utiliser une représentation similaire aux arbres de probabilité.
Créez une procédure `bern:=proc(j,n,t)` qui calcule $B_j^n(t)$.
Commentez le script suivant :

```
bezier3:=proc(p)
local Vi,Vf,B,k,j,M;
Vi:=plot([p[1],p[2]],color=green);
Vf:=plot([p[3],p[4]],color=green);
B:=p[1];
for k from 0.01 to 0.99 by 0.01 do
    M:=sum(bern(j,3,k)*p[j+1],j=0..3);
    B:=B,M;
od;
B:=plot([B,p[4]],color=blue);
plots[display](B,Vi,Vf,axes=none,scaling=constrained);
end:
```

et testez :

```
bezier3([[0,0],[1,0],[3,-3],[3,0]]);
```

Quel est le rôle des points de contrôle 2 et 3 ?

c. Avec un nombre quelconque de points de contrôle

Généralisez les procédures précédentes pour traiter un nombre quelconque de points de contrôle.
Testez avec $[[1,1],[2,3],[5,5],[6,2],[7,7],[10,5],[10,2],[8,0],[4,0],[5,1]]$.

II Courbe de Bézier du 3^e degré avec un nombre quelconque de points de contrôle

Il est pratique de travailler avec des polynômes de degré trois pour avoir droit à des points d'inflexion.
Augmenter le nombre de points de contrôle implique a priori une augmentation du degré de la fonction polynomiale.
Pour remédier à ce problème, on découpe une liste quelconque en liste de listes de 4 points.

Cependant, cela est insuffisant pour obtenir un raccordement de classe $\mathcal{C}^1$ (pourquoi ? pourquoi est-ce important d'avoir un raccordement de classe $\mathcal{C}^1$?)

Pour assurer la continuité tout court, il faut que le premier point d'un paquet soit le dernier du paquet précédent.

Le dernier « vecteur vitesse » de la liste $[P_1, P_2, P_3, P_4]$ est $\overrightarrow{P_3P_4}$. Il faut donc que ce soit le premier vecteur vitesse du paquet suivant pour assurer la continuité de la dérivée.

Appelons provisoirement le paquet suivant $[P'_1, P'_2, P'_3, P'_4]$. On a d'une part $P'_1 = P_4$ et d'autre part $\overrightarrow{P_3P_4} = \overrightarrow{P'_1P'_2}$, i.e. $P'_2 = P_4 + \overrightarrow{P_3P_4}$.

On a donc $P'_3 = P_5$ et $P'_4 = P_6$.

Connaissant `bezier3`, construire une procédure qui trace une courbe de Bézier cubique avec un nombre quelconque de points de contrôle (on prendra un nombre pair de points pour se simplifier la vie).

Testez avec $[[1,1],[2,3],[5,5],[6,2],[7,7],[10,5],[10,2],[8,0],[4,0],[5,1]]$.

Dessinez un cœur....

III B-splines uniformes

a. Cas général

Tout ceci est très beau mais il y a un hic : en changeant un point de contrôle, on modifie grandement la figure.

On considère m nœuds $t_0, t_1n, \cdots, t_m$ de l'intervalle $[0,1]$.

Introduisons une nouvelle fonction :

$$\mathbf{S}(t) = \sum_{i=0}^{m-n-1} \mathbf{P}_i b_{i,n}(t) \ , \ t \in [0,1]$$

les P_i étant les points de contrôle et les fonctions $b_{j,n}$ étant définies récursivement par

$$b_{j,0}(t) = \begin{cases} 1 & \text{si} \quad t_j \leqslant t < t_{j+1} \\ 0 & \quad \text{sinon} \end{cases}$$

et pour $n \geqslant 1$

$$b_{j,n}(t) = \frac{t - t_j}{t_{j+n} - t_j} b_{j,n-1}(t) + \frac{t_{j+n+1} - t}{t_{j+n+1} - t_{j+1}} b_{j+1,n-1}(t).$$

On ne considérera par la suite que des nœuds équidistants : ainsi on aura $t_k = \frac{k}{m}$.

On parle de B-splines uniformes et on peut simplifier la formule précédente en remarquant également des invariances par translation.

b. Fonctions B-splines d'ordre 3 avec 4 points de contrôle

À l'aide des formules précédentes, on peut prouver que dans le cas de 4 points de contrôles on obtient :

$$S(t) = \frac{1}{6}\Big((1-t)^3 P_0 + (3t^3 - 6t^2 + 4)P_1 + (-3t^3 + 3t^2 + 3t + 1)P_2 + t^3 P_3\Big)$$

Calculez $S(0)$, $S(1)$ puis $S'(0)$ et $S'(1)$: que peut-on en conclure ?

Reprenez l'étude faite avec les courbes de Bézier et comparez les résultats.