TP 4 : Interpolations - Approximations

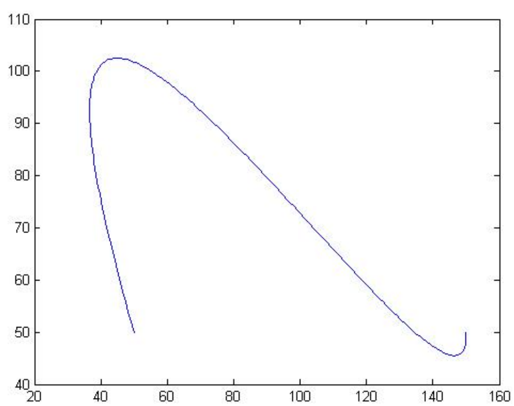
1 Cubiques de Hermite

Pour rappel, le polynôme correspondant au cubique de Hermite est le suivant :

$$P(u) = h_0(u).p_0 + h_1(u).p_1 + h_2(u).t_0 + h_3(u).t_1$$

avec : p_0 et p_1 les points de départ et de fin de la courbe, t_0 et t_1 les tangentes aux points p_0 et p_1 et les quatre polynômes h_i sont définis par :

$$\begin{aligned} h_0(u) &= (u-1)^2(2u+1) \\ h1_(u) &= u^2(3-2u) \\ h_2(u) &= u(u-1)^2 \\ h_3(u) &= u^2(u-1) \end{aligned}$$



Écrire un sous-programme sous Matlab qui permette de tracer l'interpolation par une cubique de Hermite entre les deux points p_0(50,50) et p_1(150,50) avec la tangente t_0(-180,400) en p_0 et la tangente t_1(-5,100) en p_1.

On utilisera une variable u dont la valeur progressera de 0 à 1 avec un pas de 0,01. Pour chaque valeur de u, on calculera les coordonnées du point correspondant.

On utilisera la commande **plot(X,Y)** où X donne les abscisses successives des points de la courbe et où Y donne les ordonnées. Cette commande relie chaque point successif par un segment de droite. Vous devriez obtenir le tracé ci-contre.

2 Lettres cursives vectorielles

On se propose de réaliser un code permettant de tracer les lettres de l'alphabet sous forme manuscrite. Pour cela, nous utiliserons les courbes de Bézier et une structure de données pour représenter les courbes correspondant à chaque lettre.

Rappel du polynôme de Bernstein de degrès n:

$$B_k^n(u) = C_k^n u^k (1-u)^{n-k}, \text{ avec } k \in \{0, 1, ..., n\}$$

Remarque : C_k^n peut s'écrire avec la fonction **nchoosek()** de Matlab.

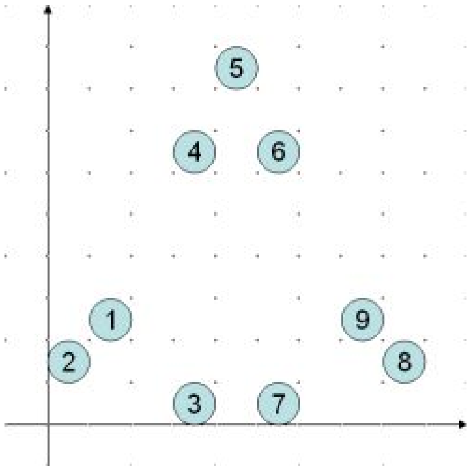
Rappel des points de De Casteljau :

$$b_k^t(u) = \sum_{j=0}^{t} b_{k+j}^0 B_j^t(u)$$

2.1 Courbe de Bézier

Écrire une fonction **bezier** qui prend comme paramètres une matrice L et une valeur réelle p et qui affiche à l'écran la courbe de Bézier décrite par ces paramètres. Le rôle de chacun d'eux est donné ci-dessous.

2.1.1 Rôle de L



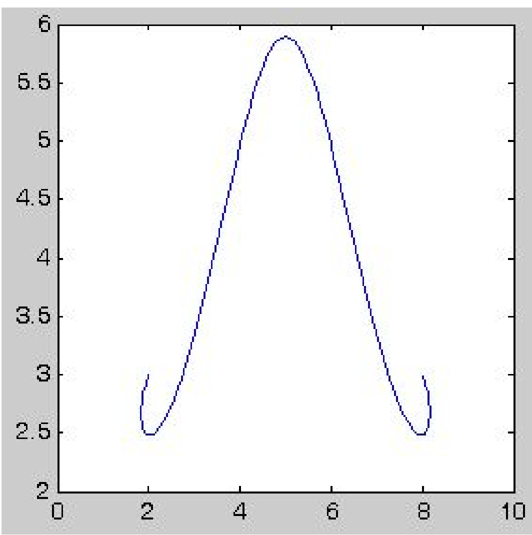
L est une matrice de 2 lignes sur n colonnes. Elle donne les coordonnées des n points du polygone de contrôles de la courbe de Bézier. La première ligne donne la succession des abscisses des points. La seconde ligne donne les ordonnées.

Exemple : La matrice L ci-dessous représente dans l'ordre les points illustrés ci-contre.

$$L = \begin{pmatrix} 2 & 1 & 4 & 4 & 5 & 6 & 6 & 9 & 8 \\ 3 & 2 & 1 & 7 & 9 & 7 & 1 & 2 & 3 \end{pmatrix}$$

On utilisera une variable u dont la valeur progressera de 0 à 1 avec un pas de 0.01. Pour chaque valeur de u, on calculera les coordonnées du point correspondant.

2.1.2 Rôle de p



p est un paramètre permettant d'ajuster la qualité de la résolution de l'interpolation. Plus exactement, p est le pas de progression du paramètre d'interpolation pour lequel les coordonnées des points successifs de la courbe vont être calculés.

Si u est le paramètre d'interpolation, on calculera les coordonnées des points pour $u = 0$, puis $u = p$, puis $u = 2p$, puis $u = 3p$... jusqu'à $u = 1$.

La figure ci-contre présente la courbe obtenue avec l'exemple donné plus haut, et pour p=0.01. Attention : sur cet exemple, les échelles des axes x et y ont été adaptées.

2.1.3 Structure de données

L'écriture d'une lettre sous forme cursive peut requérir plusieurs segments. Deux lignes sont nécessaires pour décrire les coordonnées des points correspondant à un segment. Nous utiliserons donc une matrice qui comportera deux fois plus de lignes qu'il n'y a de segments. Par exemple, la matrice ci-dessous :

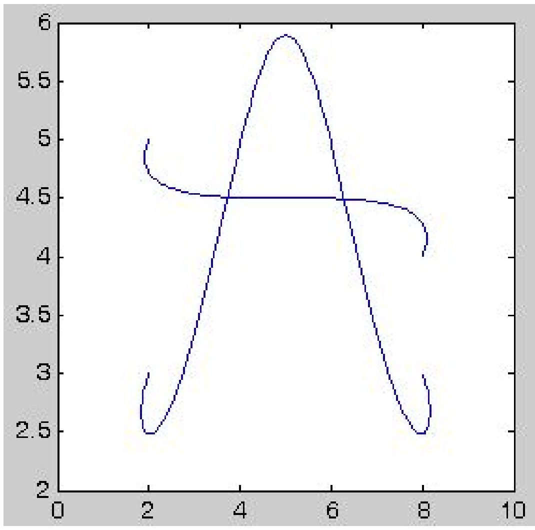
$$A = \begin{pmatrix} 2 & 1 & 4 & 4 & 5 & 6 & 6 & 9 & 8 \\ 3 & 2 & 1 & 7 & 9 & 7 & 1 & 2 & 3 \\ 2 & 1 & 9 & 8 & 0 & 0 & 0 & 0 & 0 \\ 5 & 4 & 5 & 4 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

décrit un caractère dont le tracé nécessite deux segments de courbes de Bézier. Les deux premières lignes donnent les coordonnées des points du polygone de contrôle du premier segment ; les deux dernières donnent les coordonnées des points du polygone de contrôle du second segment.

Une colonne de la matrice correspond à un point de contrôle d'un segment, ce qui suppose que tous les segments sont dessinés à partir du même nombre de points de contrôle.
Comme ce n'est pas le cas en général, nous allons formuler l'hypothèse qu'aucun point de contrôle ne se situe sur l'axe des abscisses ou des ordonnées.

En conséquence, les lignes de la matrice seront terminées par des ”zéro” dès qu’il n’y a plus de points de contrôle pour décrire le segment correspondant.

L’exemple ci-dessus illustre ce cas avec le second segment (lignes 3 et 4) : seuls quatre points de contrôle sont nécessaires pour définir le polygone. Les cinq dernières colonnes ne sont pas utilisées. Elles sont donc remplies avec des zéros.



Modifier le sous-programme **bezier** écrit précédemment de manière à pouvoir transmettre une matrice contenant les coordonnées des sommets de plusieurs polygones de contrôle, et à afficher de manière superposée toutes les courbes correspondantes.

Par exemple, le tracé défini par la matrice A définie précédemment doit être (aux facteurs d’échelle près) celui de la lettre A en majuscule cursive.

2.1.4 Alphabet cursif

Proposer sur ce principe des matrices correspondant à d’autres lettres de l’alphabet. Observer en particulier le rôle du paramètre p en réduisant sa valeur (par exemple $p = 0.2$) ou en augmentant la taille de la fenêtre contenant l’affichage graphique de la lettre.