UNIVERSITÉ de Cergy-Pontoise

Travaux dirigés sur ordinateur n°**5** : **Polynômes de Bernstein, courbes de Bézier** (↪ **Q 2**)

Question 1 : Polynôme de Bernstein

Pour un entier n fixé, on appelle $k$ième polynôme de Bernstein le polynôme

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

Pour une fonction donnée f définie sur $[0;1]$ le $n$ième polynôme de Bernstein associé à f est

$$P_n(t) = \sum_{k=0}^{n} \binom{n}{k} t^k (1-t)^{n-k} f(\frac{k}{n}) = \sum_{k=0}^{n} P(X_t = k) f(\frac{k}{n}) \text{ où } X_t \sim \mathcal{B}(n;t)$$

1. Déterminer, à la main, le polynôme de Bernstein P_2, de degré 2 associé à la fonction, $x \mapsto x^3$.
   L'instruction scilab `binomial(p,n)`, renvoie les probabilités d'une loi binomiale de paramètre n, p, par exemple `binomial(0.1,3)` renvoie la matrice $\big(P(X=0), P(X=1), P(X=2), P(X=3)\big)$ où $X \sim \mathcal{B}(3;0.1)$.
2. Dans cette question $n = 3$, $f(x) = \sqrt{x}$, à l'aide de `binomial(0.1,3)` et d'un produit matriciel calculer $P_3(0.1) = \sum_{k=0}^{3} P(X_{0.1} = k) f(\frac{k}{3})$.
3. On cherche à définir une fonction scilab `bern` qui prend en argument une matrice ligne X et qui renvoie une matrice ligne contenant les 101 éléments $\sum_{k=0}^{n} P(X_t = k) X_k$ où t prend toutes les valeurs $(0, \frac{1}{100}, \frac{2}{100}, \ldots, \frac{99}{100}, 1)$, n est égal au nombre d'éléments de X auquel on retire 1.
   Indication possible mais pas du tout obligatoire :
   (a) La fonction `length` nous permet de connaître le nombre d'éléments de X.
   (b) A l'aide d'une boucle `for`, on construit une matrice M à 101 lignes et à $n+1$ colonnes, la ième ligne étant formé de la matrice `binomial((i-1)/100,n)`.
   $$M = \begin{pmatrix} P(X_0 = 0) & P(X_0 = 1) & P(X_0 = 2) & \ldots & (X_0 = n) \\ P(X_{0.01} = 0) & P(X_{0.01} = 1) & P(X_{0.01} = 2) & \ldots & (X_{0.01} = n) \\ P(X_{0.02} = 0) & P(X_{0.02} = 1) & P(X_{0.02} = 2) & \ldots & (X_{0.02} = n) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ P(X_{0.99} = 0) & P(X_{0.99} = 1) & P(X_{0.99} = 2) & \ldots & (X_{0.99} = n) \\ P(X_1 = 0) & P(X_1 = 1) & P(X_1 = 2) & \ldots & (X_1 = n) \end{pmatrix}$$
   (c) Un produit matriciel, et deux transpositions, nous permet de calculer la matrice demandée $(\sum_{k=0}^{n} P(X_0 = k) X_k, \sum_{k=0}^{n} P(X_{0.01} = k) X_k, \sum_{k=0}^{n} P(X_{0.02} = k) X_k, \ldots, \sum_{k=0}^{n} P(X_{0.99} = k) X_k, \sum_{k=0}^{n} P(X_1 = k) X_k)$
4. Tracer sur un même dessin la fonction racine carrée sur $[0;1]$ et le quatrième polynôme de Bernstein P_3 associé à cette fonction, on pourra utiliser pour cela l'instruction `bern([0;1/3;2/3;1])`.

Question 2 : Courbes de Bézier

On remarque que $P_n(t)$ est l'espérance de $f(\frac{X_t}{n})$ où $X_t \sim \mathcal{B}(n;t)$, ceci correspond aussi au barycentre des points $f(\frac{k}{n})$ avec comme poids $P(X_t = \frac{k}{n})$. De même dans le plan en partant de $n+1$ points de coordonnées $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$, on peut prendre les barycentres de ces points munis des poids $P(X_t = \frac{k}{n})$ en faisant varier t entre 0 et 1, on obtient une courbe appelé courbe de Bézier associé aux points de contrôles $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$, ce type de courbes est très utilisée par les logiciels de dessins. Le polynôme de Bernstein était donné par $P_n(t) = \sum_{k=0}^{n} P(X_t = k) f(\frac{k}{n})$, la courbe de Bézier est paramétrée par $\varphi(t) = \big(\sum_{k=0}^{n} P(X_t = k) x_k; \sum_{k=0}^{n} P(X_t = k) y_k\big)$ la fonction `bern` nous permet de tracer très simplement des courbes de Bézier.

1. On va construire la courbe de Bézier associé aux points de contrôles suivant $A : (0;0)$, $B : (1;2)$, $C : (3;-1)$, $D : (4;0)$. Construire une matrice ligne `X` comportant les 4 abscisses et une matrice ligne `Y` comportant les 4 ordonnées.
2. Déterminer les images de `X` et `Y` par la fonction `bern` cela nous renvoie 101 barycentre des abscisses et 101 barycentre des ordonnées qu'il suffit alors de tracer avec une fonction `plot2d`.
3. Rajouter sur le dessin les 4 points de contrôle, l'utilisation de `style=-2` nous permet un tracé des 4 points.

Question 3 : L'objectif de cette question est d'automatiser le tracer de courbes de Bézier, les points de contrôle étant obtenus par un click dans la fenêtre graphique. On commencera par créer une fenêtre graphique vide de taille [-1 ;1]×[-1 ;1] à l'aide de `plot2d` et de l'option `rect`. Ensuite on construit deux vecteurs `X` et `Y` qui contiennent les abscisses et les ordonnées des points de contrôle, on programme une boucle et dans cette boucle on utilise la fonction `xclick` pour obtenir les coordonnées du nouveau point, que l'on utilise pour compléter `X` et `Y`. A chaque passage dans la boucle on trace la nouvelle courbe de Bézier.

Question 4 : Dans cette question, f est la fonction exponentielle définie sur $[0;1]$, et (P_n) la suite de ses polynômes de Bernstein.

1. Pour n variant de 5 à 30, représenter $\|\exp - P_n\|_\infty$ en fonction de n.
2. Pour n variant de 5 à 30, représenter $\ln \| \exp - P_n\|_\infty$ en fonction de $\ln n$.
3. Conjecturer l'existence d'un α tel que

$$\| \exp - P_n\|_\infty = O\left(\frac{1}{n^\alpha}\right)$$