Analyse numérique
Troisième année de licence M
2009/2010

Travaux dirigés sur ordinateur n°**2** :
Approximation uniforme par des polynômes

Question 1 : Soit f la fonction racine carrée sur $[0, 1]$.

1. Déterminer, à la main, le polynôme de Bernstein P_2, de degré 2 associé à cette fonction, on rappelle que

$$P_N = \sum_{n=0}^{N} \binom{N}{n} f\left(\frac{n}{N}\right) x^n (1-x)^{N-n}$$

2. L'instruction scilab `X=poly(0,"X")`, permet de définir par la suite des objets polynômes dont l'indéterminée est X, on crée ainsi des objets scilab, qui ne sont pas des fonctions mais des polynômes. On peut les additionner, les multiplier, et pour calculer leur valeur en un point on utilise la fonction `horner`. Par exemple l'instruction `P=X*X+X`, permet de définire le polynôme $X^2 + X$, et $horner(P, 5)$, permet de calculer sa valeur en 5, soit 30. Définir un polynôme scilab `P2` représentant le polynôme P_2 de la première question.
3. Représenter avec scilab sur un même dessin entre 0 et 1 d'une part la fonction f d'autre part P_2. on pourra par exemple définir le vecteur `x=0:0.01:1` et chercher son image par f et par P_2.

Question 2 : Aller chercher dans la page d'Alexandre Mizrahi le fichier "bernstein" suivant :

```
X=poly(0,"X")

function y=f(x)
  y=1.0./(1+x.^2)
endfunction

function P=bernstein(n)
  // Cette fonction renvoi le nième polynôme de Bernstein associé à la fonction f,
  // pour cela on construit une matrice ligne de polynômes, une matrice colonne
  // de coefficients défini avec f et enfin une matrice ligne de coefficients
  //  binomiaux. On multiplie les 3 matrices pour obtenir le polynome de Bernstein.
  pol=[]; for i=0:n, pol=[pol,X^i*(1-X)^(n-i)]; end;
  coef=[]; for i=0:n, coef=[coef;f(i/n)]; end,
  P=2^n*binomial(0.5,n).*pol*coef,
endfunction

x=[0:100]/100; y=horner(bernstein(15),x); z=f(x);
  // La fonction Horner permet d'évaluer un polynôme en un point c'est à dire
  // calculer P(x). x sont les abscisses, y leur image par le polynôme de
  // Bernstein et z leur image par f.

 clf() // permet d'effacer la page graphique de scilab
 plot2d(x,z,style=1) // style=1 permet de tracer la courbe en noir
 plot2d(x,y,style=2) // style=2 permet de tracer la courbe en bleu
```

1. Que représente la courbe bleue ? la courbe noire ?
2. Expliciter tous les calculs effectués par scilab lors de l'appel de la fonction `bernstein(2)` ?
3. On note P_n le nième polynôme de Bernstein, déterminer une valeur approchée de $\|f - P_{10}\|_\infty = \sup\{|(f - P_{10})(x)|, x \in [0; 1]\}$, on pourra utiliser la fonction `max` qui renvoie l'élément le plus grand d'une matrice. Même question avec P_{30}.

4. Tracer $f - P_{10}$ sur l'intervalle $[0;1]$, donner un argument qui laisse penser que P_{10} n'est pas la meilleur approximation uniforme de degré 10 sur $[0;1]$. On pourra regarder le nombre de fois ou $f - P_{10}$ change de signe.
5. On pose maintenant $\widetilde{f}(x) = \sqrt{|x - \frac{1}{2}|}$, montrer que $\widetilde{f}$ n'est pas Lipschitzienne, on pourra pour cela étudier la fonction dans un voisinage de $\frac{1}{2}$. Comparer $\|\widetilde{f} - \widetilde{P}_{30}\|_\infty$, avec le résultat obtenu pour la fonction f : $\|f - P_{30}\|_\infty$.

Question 3 : Dans cette question, f est la fonction exponentielle définie sur $[0;1]$, et (P_n) la suite de ses polynômes de Bernstein. , pour n variant de 5 à 30.

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

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