Master 1 Enseignement - Mathématiques **La Mi-Voix - ULCO**

ANALYSE NUMÉRIQUE - TPs SCILAB

Proposition d'exercices pour la session 1 de mai 2012

Exercice 1 (3 points) *Évaluation du polynôme d'interpolation de Lagrange par l'algorithme de Neville*

Pour évaluer le polynôme d'interpolation en un point, on peut utiliser l'algorithme de Neville que l'on rappelle ici. Cet algorithme permet en outre une estimation récursive quand on rajoute progressivement des points d'interpolation. Soit $P_{i_0\ldots,i_k}$ le polynôme d'interpolation de degré k passant par les points $(x_{i_p}, f_{i_p})_{p=0,\ldots,k}$, on établit facilement la formule récursive suivante :

$$P_{i_0\ldots,i_k} = \frac{(x - x_{i_0})P_{i_1\ldots,i_k} - (x - x_{i_k})P_{i_0\ldots,i_{k-1}}}{x_{i_k} - x_{i_0}}$$

avec

$$P_{i_\alpha} = f_{i_\alpha}.$$

Cette formule nous permet de calculer $P_n(x) = P_{0,\ldots,n}(x)$ pour une valeur de x fixée.

Programmez sous SCILAB l'algorithme de Neville pour évaluer la valeur du polynôme d'interpolation en un point x. Puis rendez-le vectoriel pour évaluer directement la valeur du polynôme pour n valeurs de x données dans un vecteur. (Utilisez la fonction $\dfrac{1}{1+x^2}$ sur $[-1, 1]$.)

Exercice 2 (2 points)

Déterminez la méthode codée par la fonction `main` suivante (implémentée sous SCILAB), en ayant au préalable détaillé ses étapes.

```
function a=f(x)
  a(1)=x(1)*(x(1)*x(1)-3*x(2)*x(2))-1 ;
  a(2)=x(2)*(3*x(1)*x(1)-x(2)*x(2)) ;
endfunction

function a=mainstep(x,h)
  n=size(x,1) ; // get the dimension of the COLUMN vector submitted
  F=f(x) ; // compute f(x)
  M=zeros(n,n) ; // prepare to make the matrix
  for i=1 :n
    v=zeros(n,1) ;
    v(i)=h ;
    M( :,i)=(f(x+v)-F)/h ;
  end
  a=x-M\F ; // do the method step
endfunction

function y=main(x,h,epsilon)
  y=x ;
  z=norm(f(y)) ; // stop when this is below epsilon
  while z>epsilon
    y=mainstep(y,h) ;
    z=norm(f(y)) ;
  end
endfunction
```