Université Abou Bekr Belkaid 2016/2017
Département de Mathématiques
Module: Analyse numérique 1
Fiche de TP N°2

Interpolation polynomiale, formule de Newton

On donne le code Matlab dividif.m fournissant les différences divisées suivant, tiré du livre: A. Quarteroni,R. Sacco et F. Saleri, Méthodes Numériques, Algorithmes, analyse et applications, Springer,2013,page 267.
Les valeurs de f aux noeuds d'interpolation x sont stockées dans le vecteur y, tandis que la matrice de sortie d (triangulaire inférieure) contient les différences divisées stockées sous la forme suivante

$$\begin{array}{llllll} x_0 & f[x_0] & & & & \\ x_1 & f[x_1] & f[x_0,x_1] & & & \\ x_2 & f[x_2] & f[x_1,x_2] & f[x_0,x_1,x_2] & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ x_n & f[x_n] & f[x_{n-1},x_n] & f[x_{n-2},x_{n-1},x_n] & \cdots & f[x_0,...,x_n] \end{array}$$

Les coefficients intervenant dans la formule de Newton sont les éléments diagonaux de la matrice.
Programme: dividif : Différences divisées de Newton

```
function [d]=dividif(x,y)
%DIVIDIF Différences divisées de Newton
% [D] = DIVIDIF(X, Y) calcule les différences divisées d'ordre n.
% X contient les noeuds d'interpolation. Y les valeurs de la fonction
% en X. D contient les différences divisée d'ordre n.
[n,m]=size(y);
if n == 1, n = m; end
n = n-1;
d = zeros (n+1,n+1);
d(:,1) = y';
for j = 2:n+1
for i = j:n+1
d (i,j) = ( d (i-1,j-1)-d (i,j-1))/(x (i-j+1)-x (i));
end
end
```

1. Après avoir enregistré ce fichier dividif.m dans votre répertoire,ouvrez le et vérifiez que la syntaxe de la fonction proposée correspond bien aux formules données en cours.
2. En vous inspirant du schèma de Horner et de la formule écrire une fonction :*function LXX = interpolN(X,Y,XX)* qui, étant donnée une liste de nombres distincts X (de longueur N+1) et une liste de nombresY de même longueur,évalue le polynôme d'interpolation aux points de la liste XX (de longueur arbitraire) et renvoie en sortie la liste LXX (de même longueur que XX) correspondant aux valeurs prises par ce polynôme d'interpolation aux entrées XX. La fonction dividif sera utilisée comme fonction auxiliaire. Sauvegardez ce fichier dans votre répertoire de travail comme interpolN.m.

3.Testez cette fonction en déclarant la fonction
f = inline('exp(-x.^2/2).*sin(pi*x)','x');
puis en déclarant :
>> X= -3:.1:3 ;
>> XX=-3:.05:3;
>> LXX=interpolN(X,f(X),XX);
>> plot(XX,f(XX),'r');
>> hold on
>> plot(XX,LXX,'k');
Qu'observez vous ? Recommencez avec cette fois les instructions
>> X= -6:.2:6 ;
>> XX=-6:.05:6 ;
>> LXX=interpolN(X,f(X),XX);
>> figure(2)
>> plot(XX,f(XX),'r');
>> hold on
>> plot(XX,LXX,'k');
Qu'observez vous cette fois ? Hors de quel segment de [-6,6] l'approximation de f par son polynôme de Lagrange se met-elle à dérailler ? Recommencez en prenant cette fois plus de points d'interpolation :
>> X=-6:.15:6;
>> XX=-6:.05:6 ;
>> LXX=interpolN(X,f(X),XX);
>> figure(3)
>> plot(XX,f(XX),'r');
>> hold
>> plot(XX,LXX,'k');
Les choses s'arrangent-elles ou empirent-elles par rapport à la figure(2) ?
Continuez avec encore plus de points :
>> X=-6:.1:6 ;
>> XX=-6:.05:6 ;
>> LXX=interpolN(X,f(X),XX);
>> figure(4)
>> plot(XX,f(XX),'r');
>> hold
>> plot(XX,LXX,'k');
Vous observez ici le phénomène de Runge. Voir pour plus de détails le site sur wikipedia :
http://fr.wikipedia.org/wiki/Ph%C3%A9nom%C3%A8ne_de_Runge