Universite Batna 2
Faculté de Technologie
Département D'hydraulique

2ème Année ST_ Hydraulique
TP Méthodes numériques S4
Année 2019/2020

TP N° 2

A.Ameziane

Interpolation et Approximation

(1-Interpolation de Newton 2- Approximation de Tchebychev)

1. Le But du TP

Dans ce TP2, nous allons tout d'abord nous intéresser aux différences divisées, à leur programmation directe et à certains avantages de cette méthode de calcul.

Ensuite, nous nous servirons de ce programme pour tracer des polynômes d'interpolation dans la base de Newton, puisque leurs coeffécients sont préecisement les différences divisées.

L'interpolation consiste à trouver l'expression générale d'une fonction à partir d'un nombre limité de points. Quand la fonction recherchée est un polynôme l'interpolation est dite polynomial.

2. Rappels

Soient n + 1 points distincts de R^2 a` interpoler : $(x_i, f(x_i))$ pour i = 0 . . . n. On rappelle que les n + 1 polynômes de Newton ω_k assocées au problème sont définis par :

- Base = $\left\{1, (x-x_0)_{,}(x-x_0)(x-x_1),\ldots\ldots(x-x_0)(x-x_1)\ldots\ldots(x-x_{n-1})\right\}$

$$w_k \in P_k : \begin{cases} w_0 = 1 \\ w_k(x) = (x-x_0)\ldots\ldots\ldots(x-x_{k-1}) = \prod_{j=0}^{k}(x-x_j) \end{cases} \qquad k = 1 \ldots n$$

Où P_k désigne l'espace vectoriel des polynômes de degré inférieur ou égal à k.

- On peut ré-ecrire P(x) :

$$P(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \cdots\ldots\ldots a_n(x-x_0)(x-x_1)\ldots(x-x_{n-1})$$

- Calcul des a_k : méthode des différences divisées

On rappelle également la formule de récurrence qui définit les n + 1 différences divisés de f :

$$\begin{cases} f[x_0] = f(x_0) \\ f[x_0, \ldots, x_k] = \frac{f[x_1,\ldots,x_k] - f[x_0,\ldots,x_{k-1}]}{x_k - x_0}, \quad k = 1 \ldots n. \end{cases} \qquad (1)$$

Le polynôme d'interpolation de f s''ecrit dans la base de Newton :

$$\prod_n f(x) = \sum_{k=0}^{n} f[x_0, \ldots, x_k] w_k(x).$$

- On considère une fonction f et deux vecteurs x et y, où les x_i sont les noeuds d'interpolation, et les $y_i = f(x_i)$ sont les valeurs de f correspondantes.
- Les différences divisées sont données par la formule (1). On peut représenter cette formule par un tableau, où le terme (i, j) est calculée à partir des eléments (i − 1, j − 1) et (i, j − 1) :

$f[x_0]$				
$f[x_1]$	$f[x_0, x_1]$			
$f[x_2]$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$		
.	.	.	.	
.	.	.	.	
.	.	.	.	
$f[x_n]$	$f[x_{n-1}, x_n]$	$f[x_{n-2}, x_{n-}, x_n]$	. . .	$f[x_0, x_1, x_2, \ldots, x_{n-1}, x_n]$.

Tableau des différences divisées

3. Questions

Question 1 : (Question 1-5 Interpolation de Newton)

On se donne les 5 points suivants dans R^2 :

x_i	−3	−1	0	2	4
f (x_i)	2	0	2	−1	1

Calculez à la main les différences divisées associées à ces 5 points.

Question 2 : Décrire l'algorithme de la procedure des différences divisées de newton

Question 3 :

- On donne l'algorithme qui permet de déterminer le polynôme d'interpolation de newton suivant :

```
Fonction a = Newton(x_i, y_i)
    pour i = 1 jusqu'à n
        F(i,0) ← y_i(i)
    fait
    pour i = 1 jusqu'à n
        pour j = 1 jusqu'à i
            F(i,j) ← (F(i,j−1) − F(i−1,j−1)) / (x_i(i) − x_i(i−j))
        fait
    fait
    pour i = 1 jusqu'à n
        a(i) ← F(n,i)
    fait
```

-

- Voici le code matlab suivant :

```
function [d,D] = coeffnewton(x, y)
% COEFFNEWTON computes the divided differences needed for
% constructing the interpolating polynomial through (x_i,y_i)
 n = length(x)-1; % degree of interpolating polynomial
 The Interpolation Polynomial 335
% divided differences
for i=1:n+1
D(i,1) = y(i);
for j = 1:i-1
 D(i,j+1) = (D(i,j)-D(i-1,j))/(x(i)-x(i-j));
end
end
d = diag(D);
```

Une fois les di calculés, pour les utiliser nous couplons avec l'algorithme de

Hörner, en ré-écrivant le polynôme p_n sous la forme :

$p_n(x) = d_0 + (x - x_0)(d_1 + (x - x_1)(d_2 + \cdots + (x - x_{n-2})(d_{n-1} + (x - x_{n-1})d_n)))$

```
function y = intnewton(x,d,z)
 % INTNEWTON evaluates the Newton interpolating polynomial
% at the new points z: y = P_n(z) using the Horner form
% and the diagonal d of the divided difference scheme.
 n = length(x)-1;
 y = d(n+1)
for i= n:-1:1
y = y.*(z-x(i))+d(i);
end;
```

- Comparer l'algorithme proposé avec le code matlab de la fonction de newton coeffnewton(x, y)

Question 4 : Programmer en langage fortran cette méthode de l'interpolation à la base de newton.

Question 5 : Posons :

$f(x) = \frac{1}{1+25x^2}$

Soient $\{x_0, x_1, \ldots, x_n\}$ les points équi-distribués sur l'intervalle [-1,1], c'est-à-dire xi =-1+2i/=n. Tracer (sur le même dessin) la fonction f et les polynômes interpolants pout quelques valeurs de n de plus en plus grandes. Qu'est-ce qu'on observe ?

Question 6 : (Approximation de Tchebychev)

Refaire les calculs du point 2 en utilisant cette fois les points de Tchebychev

$x_i = cos\frac{\pi(2i+1)}{2(n+1)}$, i=0,…, n

Qu'est-ce qu'on observe lorsque n devient de plus en plus grand ?