ESSAI - Université de Carthage
Ecole Supérieure de la Statistique et
de l'Analyse de l'Information de Tunis

1ère année
Durée : 1h30
Octobre 2011

Devoir Surveillé du module Analyse Numérique

Ines Abdeljaoued

Exercice 2 (10pt). Soient f une fonction de classe C^2 sur $[a, b]$ et $xdata = [x_0, x_1, \ldots, x_n]$ une liste de $n+1$ points deux à deux distincts de $[a, b]$.

1. Ecrire l'algorithme de la méthode de Lagrange qui prend en entrée f et $xdata$ et qui calcule le polynôme de Lagrange de f aux points $x_0, x_1, \ldots, x_n$.
2. Appliquer cet algorithme au calcul du polynôme de Lagrange pour $f = \exp(x^2)$ et $xdata = [-1, 0, 1]$ sur $-1, 1$.
3. En déduire une valeur approchée de l'intégrale $\int_{-1}^{1} \exp(x^2)\, dx$.
4. Citer trois méthodes permettant de calculer une valeur approchée de l'intégrale.

Exercice 1 (10pt). Les polynômes de Tchebychev constituent un outil important dans le domaine de l'interpolation. Soit $n \in \mathbf{N}$. Noter que pour minimiser l'erreur d'interpolation de Lagrange, les racines des polynômes de Tchebychev sont les candidats pour être les points d'interpolations. Le polynôme de Tchebychev de degré n est noté par $T_n(x) = \cos(n \arccos(x))$ pour tout $x \in [-1, 1]$ et il vérifie $T_{n+2}(x) = 2xT_{n+1}(x) - T_n(x)$ avec $T_0(x) = 1$ et $T_1(x) = x$.

1. Soit $n \in \mathbf{N}^*$. Montrer par récurrence que le coefficient du terme de degré n de T_n est 2^{n-1}.
2. Vérifier que T_n admet n racines simples définies par
$$x_k = \cos(\frac{2k+1}{2n}\pi), \quad k = 0, 1, \ldots, n-1, \quad n \in \mathbf{N}^*$$
3. Sachant que la dérivée de $T_n(x)$ est définie par $T_n'(x) = \frac{n}{\sqrt{1-x^2}} \sin(n \arccos(x))$. Montrer que T_n admet exactement $(n+1)$ extrema définis par
$$x_k' = \cos(\frac{k\pi}{n}), \quad k = 0, 1, \ldots, n.$$
En déduire que $T_n(x_k') = (-1)^k$, pour $k = 0, 1, \ldots, n$.
4. Vérifier que $\max_{x \in [-1,1]} |T_n(x)| = 1$ et en déduire que
$$\max_{x \in [-1,1]} |(x - x_0)(x - x_1) \ldots (x - x_n)| = \frac{1}{2^n}.$$

Bon Travail,
Ines Abdeljaoued.

ESSAI - Université de Carthage
Ecole Supérieure de la Statistique et
de l'Analyse de l'Information de Tunis

1ère année
Durée :1h30
Janvier 2012

Examen Analyse Numérique - Session Principale

Ines Abdeljaoued Tej

Prenez le temps de bien lire les questions. On appréciera les réponses claires et concises. Les algorithmes et les démonstrations, en particulier, doivent êtres présentés avec soin. L'épreuve comprend deux exercices.

Exercice 1 (Interpolation de Lagrange). Soit quatre points d'abscisse : $x_i = 1, 2, 3, 4$ qui ont pour ordonnées : $f_i = 2, 3, 5, 8$. L'objectif est d'effectuer une interpolation polynomiale de ces points. Pour cela on utilisera les polynômes $L_i(x)$.

1. Calculez l'expression de ces polynômes : L_i pour $i = 1, 2, 3, 4$.
2. En déduire le polynôme de Lagrange $P(x)$ qui passe par les quatre points.
3. Calculez la primitive de $P(x)$ puis l'intégrale $\int_1^4 P(x)dx$.
4. A l'aide des polynômes $L_i(x)$ calculez la fonction polynomiale $Q(x)$ qui passe par les quatre points : x_i pour $i = 0, 1, 2, 3$ et respectivement $y_i = 6, 2, 2, 6$.

Exercice 2 (Factorisation matricielle). Nous désirons écrire le programme de la décomposition LU d'une matrice carré inversible. Cet algorithme prend en entrée une matrice A carré d'ordre n tel qu'il existe L une matrice triangulaire inférieure avec des 1 sur la diagonale et U une matrice triangulaire supérieure inversible vérifiant $A = LU$. La sortie de l'algorithme est soit un message d'erreur, soit la matrice L et la matrice U.

1. Donner les conditions nécessaires et suffisantes qui assurent l'existence de la factorisation $A = LU$. Justifiez votre réponse.
2. Compléter l'algorithme de décomposition LU suivant :

```
def factLU(A):
  n = ordre de la matrice A
          L = matrice identité d'ordre n
          U = copy(A)
          si det(U)=0 alors
              return("......................................")
          sinon
              pour k allant de 0 à n-1 faire
                  si U[k,k]=0 alors
                      return('......................................')
                  sinon
                      pour i allant de 0 à n faire
                          L[i,k]=.........................
```

```
        pour i allant de k+1 à n-1 faire
            pour j allant de k+1 à n-1 faire
                U[i,j]=........................
        pour j allant de k+1 à n faire
            U[j,k]=0
return(L,U)
```

3. Appliquer l'algorithme au calcul de la factorisation LU de la matrice $A = \begin{pmatrix} -1 & 2 & -1 \\ 0 & 1 & -1 \\ -\frac{1}{2} & 2 & 0 \end{pmatrix}$.

Bon Travail.