Mathématiques pour l'informatique

LSI 1

17 - 04 - 2007

Une heure et quinze minutes, calculatrice interdite, document interdits

1 Mathématiques du signal

1. Calculer la T.F. du signal défini par $f(t) = e^{-t}H(t)$, où H désigne la fonction d'Heaviside, et en déduire, en justifiant, la T.F. du signal défini par $g(t) = t^2 e^{-t}H(t)$ (2 *pts*).
2. En déduire la valeur des intégrales
$$\int_{-\infty}^{+\infty} \frac{dx}{(1+ix)^3} \quad \text{et} \quad \int_{0}^{+\infty} \frac{dx}{(1+x^2)^3}$$
(2 *pts*).

2 Une équation récurrente

Résoudre l'équation récurrente

$$\forall n \in \mathbb{N},\ (n+1)(n+2)u_{n+2} - (n+1)u_{n+1} - 2u_n = 0 \ \text{ avec } \quad u_0 = u_1 = 1 \qquad (1)$$

(3 *pts*).

3 Algorithme de calcul de la somme d'une série alternée

On considère une fonction f positive et décroissante sur $[1, +\infty[$, vérifiant $\lim_{x\to+\infty} f(x) = 0$. On rappelle qu'alors la série
$$\sum_{n=1}^{+\infty} (-1)^n f(n)$$
est convergente (en vertu du fameux *critère des séries alternées*). Sous ces conditions, on peut également prouver que le reste $R_n = \sum_{k=n+1}^{+\infty} (-1)^k f(k)$, qui est l'erreur que l'on commet lorsqu'on calcule la somme de la série avec les n premiers termes, vérifie
$$|R_n| \leq f(n+1)$$
(autrement dit l'erreur commise est, pour une telle série **alternée**, inférieure en valeur absolue au premier terme négligé).

1. On suppose la fonction f déjà écrite :

   Fonction $f(x$: réel) : réel

   Écrire en pseudo-code une fonction *SommeSerie* qui a pour arguments en entrée la fonction f et un réel ε, et qui renvoie une valeur approchée à ε près de $\sum_{n=1}^{+\infty}(-1)^n f(n)$. Pour un coût minimal, il est conseillé de gérer les changements de signe dus au terme $(-1)^n$ en groupant les termes deux par deux (2 *pts*).
2. Les opérations élémentaires considérées sont les multiplications ou les divisions, considérées de coût équivalent. Supposons que nous souhaitions calculer une valeur approchée de $\sum_{n=1}^{\infty} \frac{(-1)^n}{n^2}$ à 10^{-4} près à l'aide de cette algorithme. Quel est le coût correspondant ? Même question si l'on souhaite calculer une valeur approchée de $\sum_{n=1}^{\infty} \frac{(-1)^n}{n^4}$ à 10^{-4} près (1,5 *pt*).
3. De façon générale, en fonction de quelle quantité devez-vous calculer le coût de cet algorithme ? Quelle "qualité" doit avoir f pour un coût raisonnable (1 *pt*) ?
4. Quelle difficulté rencontre-t-on si on veut calculer numériquement la somme d'une série (convergente bien sûr) **non alternée** (0,5 *pt*) ?

4 Principe des algorithmes modernes de factorisation

Soit p et q deux nombres premiers positifs inconnus et soit $n = pq$. On connaît n et on suppose qu'on a réussi à trouver deux entiers relatifs x et y tels que $x \neq y$ et $x \neq -y$ vérifiant

$$x^2 \equiv y^2 \mod n \tag{2}$$

Expliquer comment vous pouvez en déduire les diviseurs p et q de n, en utilisant un algorithme usuel de complexité raisonnable (2 *pts*).

Remarque. Ainsi la difficulté principale de la factorisation de n dans les algorithmes modernes (pour, par exemple, casser RSA) est la recherche de deux tels entiers x et y.