

COURS DE CALCUL FORMEL EN M1 : EXAMEN

1. UNE ÉQUATION DIFFÉRENTIELLE NON-LINÉAIRE

Le but de cet exercice est le développement en série de la solution de l'équation

$$y'(x) = x^2 - y^2(x), \quad y(0) = 1.$$

Soit $S(x)$ une série entière telle que $y(x) = S(x) + O(x^N)$ pour un entier $N \geq 1$.

(1) Montrer que $x^2 - S^2(x) - S'(x) = O(x^{N-1})$.

(2) Montrer que si $\epsilon(x)$ est solution de l'équation

$$\epsilon'(x) + 2S(x)\epsilon(x) = x^2 - S(x)^2 - S'(x), \quad \epsilon(0) = 0,$$

alors $\epsilon(x) = O(x^N)$.

(3) Montrer que cette fonction ϵ satisfait $y(x) = S(x) + \epsilon(x) + O(x^{2N+1})$.

(4) En déduire une itération calculant les 2^k premiers termes du développement en série de la solution.

(5) Si la multiplication de séries à précision m prend $O(m^2)$ opérations arithmétiques, donner une borne sur le nombre d'opérations requises pour calculer ce développement.

2. LA LEMNISCATE DE GERONO

La courbe de Viviani est l'intersection de la sphère $x^2 + y^2 + z^2 = 1$ et du cylindre $x^2 + y^2 - x = 0$. La lemniscate de Geronno est sa projection sur le plan Oyz .

(1) Montrer comment un calcul de résultant permet d'obtenir l'équation de la lemniscate. (L'équation elle-même n'est pas demandée).

(2) Pour avoir un dessin bien lisse de la courbe, il est plus commode d'utiliser une paramétrisation. Pour cela, il suffit de considérer un plan passant par l'axe Ox et dont la direction est fixée par un paramètre t . Montrer alors comment obtenir des équations paramétrées de la lemniscate de Geronno par un calcul de base de Gröbner (ces équations paramétrées ne sont pas demandées).

3. UNE FAMILLE D'INTÉGRALES

Les intégrales

$$c_{n,k} = \int_0^{+\infty} t^k K_0(t)^n dt$$

interviennent dans certains calculs physiques liés au modèle d'Ising. Dans cette équation la fonction K_0 est une fonction de Bessel qui satisfait l'équation différentielle

$$\theta^2 K_0(t) - t^2 K_0(t) = 0,$$

où θ représente l'opérateur td/dt (ainsi $\theta^2 f(t) = t(tf'(t))'$). La fonction K_0 n'a pas de singularité sur $]0, +\infty[$; aux extrémités de l'intervalle d'intégration, son comportement est donné par

$$K_0(t) \sim -\ln t, \quad t \rightarrow 0+; \quad K_0(t) \sim \sqrt{\frac{\pi}{2t}} e^{-t}, \quad t \rightarrow +\infty.$$

Le but de l'exercice est l'étude de récurrences reliant les $c_{n,k}$.

(1) Montrer que pour tout entier positif n , $K_0(t)^n$ vérifie une équation différentielle linéaire et donner un algorithme permettant de calculer une telle équation.

(2) Soit $L_0(y) = y$, $L_1(y) = \theta y$, et pour $k = 1, 2, \dots, n$

$$L_{k+1}(y) = \theta L_k(y) - k(n-k+1)t^2 L_{k-1}(y).$$

Montrer que $L_k(K_0(t)^n) = n(n-1) \cdots (n-k+1)K_0(t)^{n-k}(\theta K_0(t))^k$.

- (3) Calculer un opérateur en θ annihilant K_0^2 en appliquant cette récurrence.
(4) Montrer que pour tous entiers positifs k, j, m et n ,

$$\int_0^{+\infty} t^{k+j} \theta^m(K_0^n(t)) dt = (-1 - k - j)^m c_{n,k+j}.$$

- (5) En déduire que les $c_{2,k}$ sont reliés par la récurrence $4(k+2)c_{2,k+2} = (k+1)^3 c_{2,k}$.

4. DÉCOMPOSITION SANS CARRÉS

Un polynôme de $\mathbb{Q}[X]$ est dit *sans carré* s'il n'a que des racines simples dans \mathbb{C} . On admet que pour tout polynôme unitaire $F \in \mathbb{Q}[X]$, il existe un unique k -uplet (P_1, \dots, P_k) de polynômes sans carré, unitaires et non-constants de $\mathbb{Q}[X]$, et tels que $F = P_1 P_2^2 \cdots P_k^k$ et $\text{pgcd}(P_i, P_j) = 1$ pour $1 \leq i \neq j \leq k$. Un tel k -uplet est appelé *décomposition sans carré* de F . Voici deux algorithmes pour le calcul d'une telle décomposition.

algorithme de Tobey-Horowitz	algorithme de Yun
<p>Entrée : un polynôme unitaire F. Sortie : sa décomposition sans carré.</p> <pre> C₁ ← pgcd(F, F'); D₁ ← F/C₁; i ← 1; while C_i ≠ 1 do C_{i+1} ← pgcd(C_i, C'_i); D_{i+1} ← C_i/C_{i+1}; P_i ← D_i/D_{i+1}; i ← i + 1; end while; P_i ← C_{i-1}; return (P₁, ..., P_i) </pre>	<p>Entrée : un polynôme unitaire F. Sortie : sa décomposition sans carré.</p> <pre> C ← pgcd(F, F'); D₁ ← F/C; E₁ ← F'/C - D'₁; i ← 1; while D_i ≠ 1 do P_i ← pgcd(D_i, E_i); D_{i+1} ← D_i/P_i; E_{i+1} ← E_i/P_i - D'_{i+1}; i ← i + 1; end while; return (P₁, ..., P_{i-1}) </pre>

- (1) Prouver la correction de ces algorithmes.
(2) Évaluer et comparer leur complexité arithmétique, en fonction de $d = \deg(F)$.

5. DES SOMMES DE BINOMIAUX

- (1) En appliquant l'algorithme de Gosper, calculer

$$\sum_{k=1}^n \frac{k-1}{k(k+1)} 2^k.$$

Étant donné

$$F_{n,k} = \frac{(-1)^{n+k} k!^2 (n-k-1)!}{(n+k+1)!},$$

L'algorithme de Zeilberger permet de trouver que

$$F_{n+1,k} - F_{n,k} = G_{n,k+1} - G_{n,k}, \quad \text{où} \quad G_{n,k} = 2 \frac{n-k}{n+1} F_{n,k}.$$

- (2) En déduire la formule

$$\zeta(2) = 3 \sum_{n=1}^{\infty} \frac{1}{\binom{2n}{n} n^2}.$$