Université de Limoges
Faculté des Sciences

Année Universitaire 2008-2009

Masters Mathématiques 1
Projet de Calcul Formel : Suites Récurrentes Linéaires et Applications

L'objet de ce projet est d'étudier les suites récurentes linéaires. Ces suites interviennent, par exemple, dans le chiffrement à flot en utilisant des LFSR (registres à décalage). Les parties 1 et 2 sont à rédiger en binôme pour la fin mars. La partie pratique sera rendue en fin de semestre.

1 Préliminaires algorithmiques

1.1 Approximation rationnelle des développements limités.

Supposons que nous ayons un développement limité d'ordre $n+m+1$ d'une fonction f, c'est-à-dire $f(x) = \sum_{i=0}^{n+m} f_i x^i + \mathcal{O}(x^{n+m})$. Un *approximant rationnel de f de type* $[m,n]$ (ou encore *approximant de Padé* de type $[m,n]$) est une fraction rationnelle $\frac{P}{Q}$, avec $\deg(P) \leq m$ et $\deg(Q) \leq n$, telle que $f(x) = \frac{P}{Q}(x) + \mathcal{O}(x^{m+n+1})$.

Exercice 1

Admettons que f admette un tel approximant rationnel. Nous allons voir comment déterminer cet approximant avec l'algorithme d'Euclide étendu.
Posons $A = X^{n+m+1}$ et $B = \sum_{i=0}^{n+m} f_i X^i$. Pour simplifier les notations, nous supposons que $f_0 \neq 0$ (donc $A \wedge B = 1$). Nous commençons à appliquer l'algorithme d'Euclide étendu à A et B. Notons R_i les restes successifs dans l'algorithme d'Euclide.

1) Soit i le premier indice tel que $\deg(R_i) \leq m$ (et $\deg(R_{i-1}) > m$). Montrer que $U_i A + V_i B = R_i$ avec les relations de degré suivantes : $\deg(U_i) = n+m-\deg(R_{i-1}) < n$ et $\deg(V_i) = n+m+1-\deg(R_{i-1}) < n+1$.

2) En déduire $\frac{R_i}{V_i} = \sum_{i=0}^{n+m} f_i X^i + X^{n+m+1}\frac{U_i}{V_i}$, d'où $f - \frac{R_i}{V_i} = \mathcal{O}(X^{n+m+1})$;

3) en déduire un algorithme de calcul d'un approximant de Padé de type $[m,n]$ de f.

1.2 Division des séries

Étant donné un polynôme B, construisons une itération permettant de calculer à moindre coût un développement limité d'ordre n de $\frac{1}{B}$.

Exercice 2

1) Démontrer la proposition suivante :

Proposition 1

Soit B un polynôme de degré n dont le terme constant vaut 1. Considérons l'itération de Newton

$$\left\{ \begin{array}{rcl} g_0 & = & 1 \\ g_{k+1} & = & 2g_k - Bg_k^2 = g_k(2 - Bg_k). \end{array} \right.$$

Au bout de k itérations, il existe un polynôme r_k tel que $B.g_k = 1 + X^{2^k} r_k$.
Autrement dit, l'itération de Newton donne 2^k termes du développement limité de $\frac{1}{B}$ en k itérations.

2) Soit B un polynôme de degré n dont le terme constant vaut 1.
Notons $\mathcal{M}(N)$ le nombre d'opérations dans k nécessaires à la multiplication de deux polynômes de degré N. Nous supposons[1] que $\mathcal{M}(N) \geq 2\mathcal{M}(N/2)$.
Montrer que le calcul de $N = 2^k$ termes du développement limité de $1/B$ s'obtient en $\mathcal{N}(N) = \mathcal{O}(\mathcal{M}(N))$ opérations par l'itération de Newton ci-dessus.
(on pourra montrer , en étudiant chaque étape, que $\mathcal{N}(N) \leq 2\sum_{i=1}^{k-1} \mathcal{M}(2^{k-i})$ puis conclure en utilisant l'hypothèse sur $\mathcal{M}(N)$).

Rappel : Itération de Newton. Pour calculer une solution approchée d'une équation $\Phi(y) = 0$, on choisit $g_0 \in K$, puis on répète l'itération de Newton $g_{k+1} = g_k - \Phi(g_k)/\Phi'(g_k)$. En appliquant cette formule à $\Phi(Y) := B - 1/Y$, on retrouve l'itération ci-dessus.

2 Suites récurrentes linéaires.

2.1 Définition et propriétés

Soit K un corps effectif ; dans ce qui suit, on prendra essentiellement K de caractéristique zéro (par exemple $K = \mathbb{Q}$). On pourra ensuite considérer le cas où K est un corps fini (par exemple de caractéristique 2 pour les applications aux LFSR).

On appelle *équation de récurrence linéaire* (à coefficients constants) une équation de la forme

$$u_n = a_1 u_{n-1} + a_2 u_{n-2} + \cdots + a_{k-1} u_{n-(k-1)} + a_k u_{n-k}, \qquad (R_k)$$

où $a_1, a_2, \ldots, a_k \in K$ (avec $a_k \neq 0$). L'entier k est appelé *ordre* de l'équation.
On dit qu'une suite $(u_n)_{n\in\mathbb{N}}$ d'éléments de K est une *suite récurrente linéaire* s'il existe une équation de la forme (R_k) satisfaite par chaque (u_n) pour tout $n \geq k$. On appelle *ordre* de la suite le plus petit entier k tel que (u_n) satisfasse une récurrence linéaire (R_k) d'ordre k/.
Un exemple classique, que nous développerons ci-dessous est la récurrence de Fibonacci, définie par $F_n = F_{n-1} + F_{n-2}$.

Nous allons, dans ce qui suit, nous intéresser à deux problèmes. Le premier, étant donnée une telle suite, est de calculer le n-ième terme ; le second, étant donnés $2n$ nombres u_i, est de déterminer une suite récurrente linéaire " minimale" satisfaite par ces $2n$ nombres.

Exercice 3

On se donne une équation de récurrence linéaire (R_k). Une suite (u_n) satisfaisant (R_k) est entièrement déterminée par ses *conditions initiales* $u_0, u_1, \ldots, u_{k-1}$.

1) Montrer que l'ensemble des solutions de (R_k) est un K-espace vectoriel de dimension k.

2) On définit le *polynôme caractéristique* de la récurrence linéaire (R_k) par

$$G := X^k - a_1 X^{k-1} - a_2 X^{k-2} - \cdots - a_{k-1} X - a_k.$$

Soit λ une racine de G. Vérifier que la suite définie par $u_n = \lambda^n$ satisfait la récurrence (R_k).

3) Supposons pour l'instant que G admette n racines λ_i distinctes.Vérifier que les $(\lambda_i^n)_n$ forment une base de l'espace vectoriel des solutions. Montrer comment exprimer u_n en fonction de n, des λ_i, et des racines du polynôme caractéristique.

Exercice 4

On considère la suite de Fibonacci, définie par $F_n = F_{n-1} + F_{n-2}$ et les conditions initiales $F_0 = 0$, $F_1 = 1$. Les nombres F_i sont $0, 1, 1, 2, 3, 5, 8, 13, 21, \ldots$ Donner son polynôme caractéristique puis l'expression du terme général de F_n en fonction de n.

[1] par exemple, avec l'algorithme de Karatsuba, $\mathcal{M}(N) = \mathcal{O}(N^{1.58})$ et $\mathcal{M}(N) \geq 2\mathcal{M}(N/2)$

2.2 Calcul de u_n

Pour trouver une expression de u_n, nous proposons maintenant une meilleure méthode basée sur l'exponentiation binaire.
Réinterprétons la relation définissant u_n de manière matricielle. Nous introduisons la matrice compagnon M_G associée à la récurrence, c'est-à-dire

$$M_G = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ \vdots & & & 0 & 1 \\ a_k & a_{k-1} & \cdots & a_2 & a_1 \end{pmatrix}.$$

Alors,

$$V_{n+1} = MV_n, \quad \text{où} \quad V_n = \begin{pmatrix} u_n \\ u_{n+1} \\ \vdots \\ u_{n+k-1} \end{pmatrix}.$$

Nous avons alors $V_n = M^n V_0$, et le calcul de u_n se réduit au calcul de M^{n-k+1}.

Exercice 5

1) L'exponentiation binaire pour le calcul de A^n :
- si $n = 0$ on renvoie 1.
- si n est pair, on calcule récursivement $\tilde{A} = A^{n/2}$ puis $A^n = (\tilde{A})^2$;
- si n est impair, on calcule récursivement $\tilde{A} = A^{\frac{n-1}{2}}$ puis $A^n = A.(\tilde{A})^2$;

Montrer que cet algorithme est correct. Combien de produits de matrices $k \times k$ faut-il pour calculer A^k de cette façon ?

2) On suppose que M_G est diagonalisable. Montrer comment la méthode matricielle ci-dessus permet de retrouver les formules explicites de l'exercice précédent pour u_n.

3) On rappelle que toute matrice M est semblable (sur $\overline{K}$) à une matrice de la forme $D + N$ où:
- N est nilpotente (disons d'ordre r: $N^r = 0$),
- D est diagonale (sa diagonale est formée des valeurs propres λ_i de M),
- et $DN = ND$ (elles commutent).

Montrer que D^n s'exprime simplement, puis que $(D + N)^n$ s'exprime comme une somme de r termes (produits de puissances de D et N). En déduire une expression "explicite" de M^n en fonction des valeurs propres.

4) Retrouver le terme général de notre suite de Fibonacci en utilisant la question ci-dessus (diagonalisation de la matrice compagnon de $X^2 - X - 1$).

5) Calcul par exponentiation dans le quotient

a) Supposons que la division euclidienne donne $X^n = q_n(X)G(X) + r_n(X)$. Élaborer une méthode de type "exponentiation binaire" pour calculer X^n modulo G (attentions aux restes dans les divisions).

b) Comme $G(M_G) = \chi_{M_G}(M_G) = 0$ (par le théorème de Cayley-Hamilton[2]), nous avons $M_G^n = r_n(M_G)$.En déduire une nouvelle méthode de calcul de M_G^n.

c) Pour notre suite de Fibonacci, le polynôme caractéristique est $G = X^2 - X - 1$. Supposons que $X^{2^m} \equiv a_m X + b_m \mod G$. Déterminer les formules de récurrence satisfaites par les a_m,b_m.

[2]on vérifie aisément que le polynôme caractéristique χ_{M_G} de M_G est le polynôme caractéristique G de la récurrence (ça tombe bien)

2.3 Calculer tous les u_n, ou aussi retrouver le polynôme caractéristique

Considérons maintenant le problème suivant : étant donnés $2n$ nombres u_n, trouver une suite récurrente linéaire (R_k) d'ordre n satisfaite par ces nombres. La technique que nous introduisons pour ce problème donnera aussi une méthode alternative pour calculer les u_n (si on les veut tous).
Définition. On appelle *série génératrice* de la suite (u_n) l'expression (formelle)

$$\hat{f} = \sum_{i=0}^{\infty} u_i X^i.$$

La " série formelle" ci-dessus peut être pour l'instant pensée comme un objet vérifiant $\hat{f}(x) = \sum_{i=0}^{n} u_i x^i + \mathcal{O}(x^{n+1})$ pour tout n.
Pour un polynôme G de degré n, on appelle *polynôme réciproque* de G le polynôme $H(X) := X^n G(\frac{1}{X})$.

Exercice 6

1) Montrer la proposition suivante:

Proposition 2

La suite (u_n) est une suite récurrente linéaire d'ordre k si et seulement si sa série génératrice est rationnelle, c'est-à-dire s'il existe des polynômes F, H de degré (au plus) k tels que : $H(0) = 1$ et, pour tout $n > 2k$, $\frac{F}{H} - \sum_{i=0}^{n} u_i X^i = \mathcal{O}(X^{n+1})$. De plus, le polynôme H est le polynôme réciproque du polynôme caractéristique G de la suite (u_n).

2) Déterminer l'approximant rationnel (ou approximant de Padé) de la série génératrice de la suite de Fibonacci étudiée dans les exemples ci-dessus.

3) Proposer un algorithme qui, étant donné $2n + 1$ nombre $(u_0, \ldots, u_{2n})$ détermine (sous l'hypothèse que la suite est récurrente linéaire d'ordre au plus n) un approximant rationnel de leur série génératrice, puis le polynôme caractéristique de cette (éventuelle) suite récurrente linéaire.

4) Développer, étant donné un approximant rationnel, une méthode " à la Newton" de calcul de n termes u_i de la série (on pourra supposer, pour simplifier, que $n = 2^k$ dans un premier temps).

3 Partie pratique du projet

Programmer les algorithmes ci-dessus et comparer les différentes stratégies proposées (sauf 5.2 et 5.3, utilisant diagonalisation et décomposition de Dunford, qui sont facultatives car délicates à programmer). Comparer la complexité expérimentale et la complexité pratique, proposer des exemples amusants, etc.

Un exemple d'application: Les *registres à décalage à rétroaction linéaires* (Linear Feedback Shift Registers, LFSR) de longueur n sont des dispositifs utilisés en codage et cryptographie. Le principe est le suivant: le bit de droite est le résultat (la " sortie" du registre) et, à chaque top d'horloge, le nouveau bit est donné par

$$s_{i+n} = c_1 s_{i+n-1} + c_2 s_{i+n-2} + \cdots + c_n s_i,$$

où les $c_i \in \{0, 1\}$ sont des coefficients binaires.
Si on travaille sur F_2, on peut les utiliser pour faire du chiffrement à flot (en étant convenus d'un générateur d'état initial, par exemple le jour de l'envoi). Il faut alors s'arranger pour avoir une périodicité suffisament grande pour qu'un espion Oscar ne puisse pas calculer le polynôme de rétroaction.

Si l'on ne connaît pas les c_i mais qu'on mesure $2n$ valeurs des s_i, le procédé d'approximation rationnelle ci-dessus permet de retrouver le polynôme caractéristique de la suite. La série $\sum s_i x^i$ est rationelle. On a $\sum s_i x^i = \frac{p(x)}{g(x)}$ où g donne le polynôme caractéristique et p donne l'état initial de la suite.
C'est une variante (un peu simplifiée) d'un algorithme de Berlekamp-Massey[3].

[3] Algorithme dû à Elwyn Berlekamp en 1968. Sa connection avec les codes linéaires a été observée par James Massey en 1969.