CALCULATRICES ET CALCUL LITTERAL

Jean SAMSON & Raymond SEROUL

0. INTRODUCTION

Je voudrais développer le polynôme

$$P(X) = (X+1)^4(3X^2-6X+7) + (9X^3-3X^2+12X-11)^2 - 5(X+31)^5.$$

Je dispose d'une calculatrice programmable : puis-je lui confier ce calcul ? Cette question, posée tout de go à une personne qui n'a pas réfléchi au problème amène en général la réponse suivante : "comment une calculatrice, qui ne peut manipuler que des nombres décimaux, pourrait-elle effectuer des calculs portant sur une entité abstraite telle un polynôme !".

Une première réflexion montre que cette objection n'est pas fondée. En effet, la notation aX^2+bX+c n'est qu'un moyen mnémotechnique (extraordinaire, il faut bien l'avouer) pour manipuler le triplet (a,b,c) selon certaines lois. Par conséquent, une calculatrice doit être capable de s'occuper de ce calcul. Reste à trouver une méthode.

Que peut donc faire une calculatrice ? Si je lui fournis un nombre réel x, elle peut calculer le nombre $P(x)$. La question devient donc : peut-on reconstituer le polynôme $P(X)$ à partir d'un nombre fini de ses valeurs ? La réponse est donnée par le théorème que voici.

1. LA THEORIE DU POLYNOME D'INTERPOLATION DE LAGRANGE

THEOREME 1. **Soient $x_0, x_1, \ldots, x_n$ n+1 nombres distincts. Etant donnés $y_0, y_1, \ldots, y_n$ quelconques, il existe un unique polynôme $L(X)$ de degré $\leqslant n$ qui satisfait les n+1 égalités $L(x_i) = y_i$.**
Cet unique polynôme porte le nom de polynôme d'interpolation de LAGRANGE.

DEMONSTRATION : Il y a unicité, car deux polynômes de degré $\leqslant n$ et qui coïncident en n+1 points distincts sont égaux. Prouvons maintenant l'existence. Posons

$$\omega(X) = (X-x_0)\ldots(X-x_n), \quad \omega_k(X) = \omega(X)/(X-x_k) \quad 0 \leqslant k \leqslant n.$$

Le polynôme $\omega_k(X) / \omega_k(x_k)$ est de degré n et vérifie les égalités $\frac{\omega_k(x_i)}{\omega_k(x_k)} = \delta_{ki}$. Par conséqu nt, le polynôme

$$(1) \qquad L(X) = \sum_0^n y_k \, \omega_k(X) / \omega_k(x_k)$$

répond à la question.

La première idée qui vient à l'esprit consiste à utiliser la formule (1) pour reconstituer le polynôme P(X) à partir de ses valeurs. L'inconvénient est que cette formule est très coûteuse du point de vue des calculs. Par bonheur, il existe une autre présentation du polynôme d'interpolation de LAGRANGE, présentation parfaitement adaptée aux calculs et qui est dûe à NEWTON. N'oublions pas qu'à cette époque les calculs se faisaient entièrement à la main ! Voici donc la version de NEWTON.

Soit P(X) un polynôme de degré n. Donnons-nous n+1 nombres **quelconques** $x_0,\dots,x_n$. Si nous posons

$$P(x_0,X) = (\, P(X)-P(x_0) \,)/(X-x_0)$$

nous définissons un polynôme $P(x_0,X)$ de degré n-1 qui vérifie l'identité

$$P(X) = P(x_0) + P(x_0,X)(X-x_0).$$

Recommençons en posant

$$P(x_0,x_1,X) = P(x_0,x_1) \;/\; (X-x_1).$$

Cela définit un polynôme $P(x_0,x_1,X)$ de degré n-2 qui vérifie l'identité

$$P(X) = P(x_0) + P(x_0,x_1)(X-x_0) + P(x_0,x_1,X)(X-x_0)(X-x_1).$$

Nous pouvons continuer ainsi jusqu'au polynôme constant

$$P(x_0,\dots,x_{n-1},X) = P(x_0,\dots,x_{n-1},x_n)$$

et obtenir l'identité

$$(2) \qquad P(X) = \sum_0^n P(x_0,\dots,x_k)\, C_k(X)$$

où l'on a posé, pour simplifier,

$$C_0(X) = 1,\; C_1(X) = X-x_0,\dots,C_n(X) = (X-x_0)\dots(X-x_{n-1}).$$

Les coordonnées $P(x_0,\dots,x_k)$ du polynôme P(X) dans la base des $C_k(X)$ s'appellent les **différences divisées** du polynôme P(X). La question qui se pose maintenant est : peut-on trouver un algorithme qui permette de calculer ces différences divisées et qui soit plus économique que la détermination des polynômes $P(x_0,X), P(x_0,x_1,X)\dots$

Si l'on suppose $x_0,\dots,x_n$ **deux à deux distincts**, la réponse est immédiate. En effet, la définition même des polynômes $P(x_0,\dots,x_k,X)$ montre que l'on a les égalités

(3)
$$P(x_0,x_k) = (P(x_k)-P(x_0))/(x_k-x_0) \qquad 1 \leqslant k \leqslant n$$
$$P(x_0,x_1,x_k) = (P(x_0,x_k)-P(x_0,x_1))/(x_k-x_1) \qquad 2 \leqslant k \leqslant n$$
et plus généralement
$$P(x_0,\dots,x_i,x_k)=(P(x_0,\dots,x_{i-1},x_k)-P(x_0,\dots,x_{i-1},x_i))/(x_k-x_i)$$
$$i < k \leqslant n.$$

Lorsque les x_k ne sont pas tous distincts, les formules précédentes ne s'appliquent pas toujours. Aussi aurons-nous besoin des résultats que voici.

PROPOSITION 2. **Pour tout entier $0 \leqslant k \leqslant n$ et pour tout nombre réel x on a**

(4) $$P(x(k+1)) = p^{(k)}(x)/k!$$

où l'on a posé pour simplifier $x(k+1) = (x,\dots,x)$ (k+1 fois).

DEMONSTRATION. Posons $A = (x_0,\dots,x_{k-1})$ et $x = x_k$. Par définition même, nous avons l'identité
$$P(A,X) = P(A,x) + P(A,x,X)(X-x).$$
Dérivons k fois par rapport à X. La formule de Leibniz montre que l'on a
$$P(A,X)^{(k)} = P(A,x,X)^{(k)}(X-x) + k\,P(A,x,X)^{(k-1)}.$$
(attention : ne pas confondre la dérivée k-ième de $P(A,X)$ avec le polynôme $P^{(k)}(A,X)$ obtenu à partir de $P^{(k)}(X)$). Cela nous donne la congruence
$$P(A,X)^{(k)} \equiv k\,P(A,x,X)^{(k-1)} \qquad \text{mod } X-x.$$
Nous avons donc
$$P(X)^{(k)} \equiv k\,P(x,X)^{(k-1)} \equiv k(k-1)\,P(x,x,X)^{(k-2)} \equiv \dots$$
d'où finalement la congruence
$$P(X)^{(k)} \equiv k!P(x(k+1)) \qquad \text{mod } X-x.$$
Il ne reste plus qu'à faire X=x pour obtenir le résultat désiré.

PROPOSITION 3. **Pour tout entier $0 \leqslant k \leqslant n$ et toute permutation s on a l'égalité**

(5) $$P(x_{s(o)},\dots,x_{s(n)}) = P(x_o,\dots,x_n).$$

DEMONSTRATION. Il suffit de se borner au cas où s est l'une des transpositions $(0,1),(1,2),\ldots,(k-1,k)$. L'identité

$$P(x_{s(o)},\ldots,x_{s(k-1)},X) = P(x_0,\ldots,x_{k-1},X)$$

se démontre facilement par récurrence à partir de l'identité

$$P(A,x,y,X) = \left[(y-x)P(A,X)+X(P(A,x)-P(A,y))+x\,P(A,y)-y\,P(A,x)\right] / (y-x)(X-x)(X-y)$$

identité obtenue en appliquant deux fois de suite (3). Par conséquent, l'égalité (5) est vraie lorsque $s \neq (k-1,k)$. Si l'on a $s = (k-1,k)$, l'égalité (5) est une conséquence immédiate de (3).

COROLLAIRE 4. **Si l'on a $x \neq y$, on a (avec des notations évidentes) l'égalité**

$$(6) \qquad P(A,x,B,y,C) = \left[P(A,B,y,C) - P(A,x,B,C)\right] /(y-x).$$

En effet, on a $P(A,x,B,y,C) = P(A,B,C,x,y)$. Il ne reste plus qu'à utiliser (5).

2. UN ALGORITHME POUR DÉVELOPPER UN POLYNÔME A UNE INDETERMINÉE

Soit $P(X)$ le polynôme à développer. Choisissons arbitrairement $n+1$ nombres distincts $x_0,\ldots,x_n$, où n désigne le degré de $P(X)$. En nous appuyant sur (2) et (3) nous obtenons l'algorithme suivant :

- première étape : calcul des $y_k = P(x_k)$
- deuxième étape : calcul des différences divisées $P(x_0,\ldots,x_k)$ $0 \leqslant k \leqslant n$, au moyen des égalités (3).
- troisième étape : on exprime dans la base canonique $1,X,\ldots,X^n$ le polynôme d'interpolation (2).

Précisons un peu ce que va être cette troisième étape. A la fin de la deuxième étape , le polynôme $P(X)$ est exprimé dans la base des $C_k(X)$. Nous allons successivement exprimer dans la base canonique les polynômes

$$P_0 = P(x_o)C_o,\ P_1 = P_o + P(x_o,x_1)C_1,\ldots,\ P_n = P_{n-1} + P(x_o,\ldots,x_n)C_n.$$

Pour cela, supposons connus les coefficients des polynômes C_{k-1} et P_{k-1}, soit

$$C_{k-1} = \sum_0^n c_i X^i \quad \text{et} \quad P_{k-1} = \sum_0^n a_i X^i.$$

Puisque nous avons $C_k = C_{k-1}(X-x_{k-1})$ pour $1 \leqslant k \leqslant n$, nous aurons donc

$$C_k = -x_{k-1}c_o + \sum_1^n (c_{i-1}-c_i x_{k-1})X^i$$

$$P_k = \sum_0^n (a_i + P(x_0,\dots,x_k)c_i)\ X^i.$$

Algorithme pour développer un polynôme à une indéterminée.

Cet algorithme permet d'exprimer dans la base canonique un polynôme dont on connaît les valeurs en n+1 points. Il nécessite 4(n+1) mémoires $x_o,\dots,x_n, y_o,\dots y_n, c_o,\dots,c_n\ \ a_o,\dots,a_n$.

0) (initialisation) données : $n \geqslant 1$ et $x_o,\dots,x_n$ deux à deux distincts.

1) (calcul et mise en mémoire des $P(x_k)$)

faire $y_o = P(x_o), y_1=P(x_1),\dots,y_n = P(x_n)$.

2) (calcul des différences divisées selon (3). A la fin de ce numéro, on a $y_o = P(x_o), y_1 = P(x_o,x_1),\dots,y_n = P(x_0,\dots,x_n)$).

```
pour i = 1,...,n faire
|  pour j = i,...,n faire
|  |  y_j = (y_j-y_{i-1})/(x_j-x_{i-1})
|  refaire
refaire
```

3) (on connaît maintenant le polynôme $P(X)$ dans la base des C_k. Pour $k = 0,\dots,n$ on exprime les polynômes C_k et P_k dans la base canonique)

(initialisation $P_o = P(x_o)$, $C_o = 1$)

faire $a_o = y_o,\ a_1 = 0,\ \dots,\ a_n = 0$

$c_o = 1\ ,\ c_1 = 0,\ \dots,\ c_n = 0$

(changement de base)

```
pour k = 1,...,n faire
|  pour i = n,...,1 (dans cet ordre !) faire
|  |  c_i = c_{i-1} - c_i x_{k-1}
|  |  a_i = a_i + y_k c_i
|  refaire
|  faire
|     c_o = -x_{k-1} c_o
|     a_o = a_o + y_k c_o
refaire
```

4) le polynôme $P(X)$ est le polynôme $\sum_0^n a_i\ X^i$.

3. UN TRES JOLI THEORÊME

Si $y_o, \ldots, y_n$ sont des nombres arbitraires, les parties 2) et 3) de l'algorithme que nous venons de décrire déterminent un polynôme $P(X)$ tel que l'on ait $P(x_k) = y_k$. Si les x_k ainsi que les y_k sont des entiers, il n'est pas vrai que le polynôme $P(X)$ soit à coefficients entiers. Précisons cela. Faisons le choix

$$x_o = 0,\ x_1 = 1,\ \ldots,\ x_n = n.$$

Si nous posons

$$\Delta P(X) = P(X+1) - P(X)$$
$$\Delta^2 P(X) = \Delta(\Delta P(X)) = P(X+2) - 2P(X+1) + P(X)$$
$$\ldots\ldots$$

une récurrence à l'aide de (6) montre facilement que l'on a

$$P(0,1,\ldots,k) = \Delta^k P(0)\ /\ k!.$$

L'identité (2) se réécrit donc

$$(7) \qquad P(X) = P(0) + \Delta P(0)X/1! + \Delta^2 P(0)X(X-1)/2! + \ldots + \Delta^n P(0)X(X-1)\ldots(X-n+1)/n!.$$

Cette identité va nous permettre de démontrer le théorème suivant.

THEOREME 5. **On a l'inclusion $P(\mathbb{Z}) \subset \mathbb{Z}$ (c'est-à-dire P prend des valeurs entières sur les entiers) si et seulement si $P(0), \Delta P(0), \ldots, \Delta^n P(0)$ sont tous des entiers.**

La démonstration est immédiate à partir de (7) et des remarques suivantes :

- si P est à valeurs entieres, il en est de même des polynômes $\Delta P,\ \Delta^2 P, \ldots,\ \Delta^n P$.
- les polynômes $C_k(X)/k!$ prennent des valeurs entières sur les entiers (penser aux coefficients du binôme).

Remarque : on peut éviter de faire appel à la théorie du polynôme d'interpolation de LAGRANGE pour obtenir l'identité (7). Pour cela, on écrit le polynôme P dans la base des C_k, soit $P = d_o + d_1 C_1 + \ldots + d_n C_n$. Il est clair que l'on a $d_o = P(O)$. Ensuite, on remarque que pour $k \geqslant 1$, on a $\Delta C_k = C_{k-1}$. Par conséquent, on a $\Delta P = d_1 + d_2 C_1 + \ldots + d_n C_{n-1}$, d'où $d_1 = \Delta P(0)$, etc.

4. LA THEORIE DU POLYNOME D'INTERPOLATION DE HERMITE

Le problème de l'interpolation à la HERMITTE est le suivant. On se donne des nombres réels $x_0, \ldots, x_r$ deux à deux distincts ainsi que des entiers $m_o, \ldots, m_r$. Existe-t-il un polynôme $H(X)$, de degré $\leqslant n$, avec $n+1 = (m_o+1)+\ldots+(m_r+1)$, et tel que l'on ait les égalités

$$(8)\quad \begin{cases} H(x_o) = y_o^{(o)}\ , \ H'(x_o) = y_o^{(1)}\ , \ \ldots\ , \ H^{(m_o)}(x_o) = y_o^{(m_o)} \\ H(x_r) = y_r^{(o)}\ , \ H'(x_r) = y_r^{(1)}\ , \ \ldots\ , \ H^{(m_o)}(x_r) = y_r^{(m_o)}. \end{cases}$$

étant entendu que les $y_i^{(j)}$ sont eux aussi donnés à l'avance.

THEOREME 6. **Il existe un polynôme H(X) et un seul qui soit de degré $\leqslant n$ et qui vérifie les égalités (8).**

DEMONSTRATION. Il y a unicité, car la différence entre deux solutions est divisible par $(X-x_o)^{m_o+1}(X-x_1)^{m_1+1}$. Prouvons alors l'existence d'une solution. Pour cela, considérons la famille de polynômes

$$\psi_k(X) = (X-x_o)^k(X-x_1)^{m_1+1}\ldots(X-x_r)^{m_r+1}.$$

Puisque $x_1,\ldots,x_r$ sont des zéros d'ordre de multiplicité $m_1+1,\ldots m_r+1$ du polynôme ψ_k il en résulte que l'on a $\psi_k^{(i)}(x_j) = 0$ pour $1 \leqslant j \leqslant r$ et $0 \leqslant i \leqslant m_j$.

D'autre part, nous avons

$$\psi_o(x_o) \neq 0$$

$$\psi_1(x_o) = 0,\ \psi_1'(x_o) \neq 0$$

$$\psi_k(x_o) = 0,\ \psi_k'(x_o) = 0,\ \ldots,\ \psi_k^{(k-1)}(x_o) = 0,\ \psi_k^{(k)}(x_o) \neq 0.$$

Il est donc possible de trouver des constantes $a_o,\ldots,a_{m_o}$ telles que le polynôme

$$H_o(X) = a_o\,\psi_o(X) + \ldots + a_{m_o}\,\psi_{m_o}(X)$$

satisfasse la première ligne des conditions (8).

On détermine ensuite de manière analogue des polynômes $H_1,\ldots,H_r$ qui satisfont les autres lignes des conditions (8). Il est alors clair que le polynôme

$$(9)\quad H(X) = H_o(X) + \ldots + H_r(X)$$

est une solution.

Cet unique polynôme porte le nom de polynôme d'interpolation de HERMITE.

Bien entendu, la formule (9) n'est qu'une formule théorique. Il est hors de question de l'utiliser dans un calcul pratique.

Montrons sur un exemple comment on peut procéder pour déterminer le polynôme H(X) de manière économique.

Soit à déterminer le polynôme H(X) de degré $\leqslant 7$ qui satisfait les conditions

$$\begin{cases} H(0) = 1 & H'(0) = 0 & H''(0) = 0 \\ H(1) = 2 & H'(1) = 7 & \\ H(2) = 129 & H'(2) = 448 & H''(2) = 1344 \end{cases}$$

En vertu de l'identité (2), nous pouvons écrire

$$\begin{aligned} H(X) = H(0) &+ H(0,0)X + H(0,0,0)X^2 + H(0,0,0,1)X^3 \\ &+ H(0,0,0,1,1)X^3(X-1) + H(0,0,0,1,1,2)X^3(X-1)^2 \\ &+ H(0,0,0,1,1,2,2)X^3(X-1)^2(X-2) + H(0,0,0,1,1,2,2,2)X^3(X-1)^2(X-2)^2. \end{aligned}$$

Considérons le tableau suivant (où (0012) désigne H(0,0,1,2)) :

x	y	L=0	L=1	L=2	L=3	L=4	L=5
0	1	(0)=1 ←	(0)=1	(0)=1	(0)=1	(0)=1	(0)=1
0	0	(00)=0	(00)=0 ←	(00)=0	(00)=0	(00)=0	(00)=0
0	0	(000)=0	(000)=0	(000)=0 ←	(000)=0	(000)=0	(000)=0
1	2	(1)=2	(01)=1	(001)=1	(0001)=1 ←	(0001)=1	(0001)=1
1	7	(11)=7	(011)=6	(0011)=5	(00011)=4	(00011)=4 ←	(00011)=4
2	129	(2)=129	(02)=64	(002)=32	(0002)=16	(00012)=15	(000112)=11 ←
2	448	(22)=448	(022)=192	(0022)=80	(00022)=32	(000122)=17	(0001122)=6
2	1344	(222)=672	(0222)=240	(00222)=80	(000222)=24	(0001222)=7	(00011222)=1

Les deux premières colonnes de ce tableau sont les données (les x_k et les y_k).

La colonne L=0 s'obtient à partir des données en utilisant (4).

Pour passer de la colonne L=0 à la colonne L=1 on procède comme suit :

- on ne fait rien dans le bloc qui contient la flèche.
- le premier élément de chaque bloc se calcule à l'aide de (6), soit

  $H(0,1) = (H(1)-H(0))/(1-0) = (2-1)/(1-0) = 1$

  $H(0,2) = (H(2)-H(0))/(2-0) = (129-1)/(2-0) = 64.$
- une fois calculé le premier élément d'un bloc, les autres éléments de ce bloc se calcule par récurrence à partir du premier élément en utilisant toujours (6), ce qui donne

$H(0,1,1) = (H(1,1)-H(0,1))/(1-0) = (7-1)/(1-0) = 6$

$H(0,2,2) = (H(2,2)-H(0,2))/(2-0) = (44-64)/(2-0) = 192$

$H(0,2,2,2) = (H(2,2,2)-H(0,2,2))/(2-0) = (672-192)/(2-0) = 240.$

On passe de manière analogue aux colonnes suivantes. La colonne L=5 fournit les différences divisées requises. Le polynôme cherché est donc

$$H(X) = 1+1.X^3+4.X^3(X-1)+11.X^3(X-1)^2+6.X^3(X-1)^2(X-2) + 1.X^3(X-1)^2(X-2)^2$$
$$= 1 + X^7.$$

Algorithme pour déterminer le polynôme d'interpolation de HERMITE

Cet algorithme nécessite les mêmes mémoires que l'algorithme de la partie 2.

0) (initialisation) les données sont stockées dans les mémoires $x_o,\dots x_n$ et $y_o,\dots,y_n$ **dans l'ordre suivant**

$$x_o, \dots, x_o \quad , \dots, x_r \ , \dots, x_r$$
$$y_o^{(o)}, \dots, y^{(m_o)}/m_o! \ , \dots, y_r^{(o)}, \dots, y_r^{(m_r)}/m_r!$$

1) (calcul des différences divisées : à la fin de ce numéro, on a $y_o=H(x_o)$, $y_1 = H(x_o,x_o),\dots,y_n = H(x_o,\dots,x_r)$)
(l'indice 1 correspond aux flèches dans le tableau, l'indice i pointe sucessivement sur le premier élément de chaque bloc)

```
pour l = 0, ..., n-1 faire
   faire i = l
   pour k = l+1, ..., n faire
   |  tant que x_k ≠ x_l faire
   |  |  si x_k ≠ x_i alors faire i = k fin si
   |  |  si k = i alors faire y_k = (y_k-y_l)/(x_k-x_l)
   |  |  |  sinon faire       y_k = (y_k-y_{k-1})/(x_k-x_l)
   |  |  fin si
   |  refaire
   refaire
```

2) (on connaît maintenant le polynôme H dans la base des C_k. On exprime les polynômes C_k et P_k dans la base canonique)
reprendre ici le numéro 3) de la partie 2

3) le polynôme H cherché est le polynôme $\sum_0^n a_i X^i$.

5. CAS D'UN POLYNOME A DEUX INDETERMINEES

Supposons maintenant que je veuille développer le polynôme

$$Q(X,Y) = (3X^3+5X^2Y-12XY^2-17Y^3)(X-4Y+9)(X+Y-1) - (X^5-11X^3+Y^2)(X-8Y+3).$$

L'algorithme exposé dans la partie 2 se généralise aisément au cas de deux indéterminées.

Soit $Q(X,Y)$ un polynôme de degré n en X et de degré m en Y. Donnons-nous

$x_o, \ldots, x_n$ deux à deux distincts

$y_o, \ldots, y_m$ deux à deux distincts.

Le théorème que voici est une conséquence immédiate du théorème 1.

THEOREME 7. **Etant donnée une famille z_{ij} $0 \leqslant i \leqslant n$, $0 \leqslant j \leqslant m$ de nombres réels, il existe un polynôme et un seul $Q(X,Y)$, de degré $\leqslant n$ en X et $\leqslant m$ en Y, vérifie les égalités $Q(x_i,y_j) = z_{ij}$.**

Montrons maintenant comment on détermine explicitement ce polynôme. En procèdant de manière analogue à la partie 1 nous pouvons définir des polynômes $Q(X;y_o,\ldots,y_j,Y)$ en travaillant dans l'anneau $\mathbb{R}[X][Y]$.

Ces polynômes permettent de démontrer l'identité

$$Q(X,Y) = \sum_{1}^{m} Q(X;y_o,\ldots,y_j)D_j(Y)$$

où l'on a posé

$$D_o(Y) = 1,\ D_1(Y) = Y-y_o, \ldots,\ D_m(Y) = (Y-y_o)(\ldots(Y-y_m).$$

La différence divisée $Q(X;y_o,\ldots,y_j)$ est un polynôme en X de degré $\leqslant n$. On peut donc considérer les différences divisées $Q(x_o,\ldots,x_i;y_o,\ldots,y_j)$ de ce polynôme, ce qui amène l'identité

$$Q(X;y_o,\ldots,y_j) = \sum_{1}^{n} Q(x_o,\ldots,x_i;y_o,\ldots,y_j)\ C_i(X).$$

En combinant entre elles les identités obtenues, nous obtenons l'identité

$$(10) \quad Q(X,Y) = \sum_{i,j} Q(x_o,\ldots,x_i;y_o,\ldots,y_j)\ C_i(X)\ D_j(Y).$$

Remarquons que l'unicité de la décomposition de $Q(X,Y)$ dans la base des $C_i(X)D_j(Y)$ montre que l'on aurait pu échanger le rôle des variables X et Y sans changer le résultat final.

Pour développer le polynôme $Q(X,Y)$ nous procèderons comme suit :

- première étape : calcul des $z_{ij} = Q(x_i,y_j)$
- deuxième étape : calcul des différences divisées $Q(x_o,\ldots,x_i;$ $y_o,\ldots,y_j)$
- troisième étape : expression du polynôme obtenu en (10) dans la base canonique des X^iY^j.

La deuxième étape comportera deux passes :

- première passe : calcul des $Q(x_o,\dots,x_i;y_j)$
- deuxième passe : calcul des $Q(x_o,\dots,x_i;y_o,\dots,y_j)$.

Précisons aussi ce que sera la troisième étape.

Elle comportera elle aussi deux passes de calculs. Si nous posons

$$A_j(X) = Q(X;y_o,\dots,y_j) = \sum_i Q(x_o,\dots,x_i;y_o,\dots,y_j)\ C_i(X)$$

l'identité (10) s'écrit

$$Q(X,Y) = \sum_j A_j(X)\ D_j(Y).$$

La première passe consistera à décomposer chaque polynôme $A_j(X)$ dans dans la base canonique $1,\dots,X^n$. Dans ce but, nous exprimerons successivement les polynômmes

$$A_{o,j} = Q(x_o;y_o,\dots,y_j)\ C_o(X)$$

$$A_{1,j} = A_{o,j} + Q(x_o,x_1;y_o,\dots,y_j)\ C_1(X)$$

$$\vdots$$

$$A_{n,j} = A_{n-1,j} + Q(x_o,\dots,x_n;y_o,\dots,y_j)\ C_n(X) = A_j(X)$$

dans la base canonique. Si nous supposons connus les polynômes C_{k-1} et $A_{k-1,j}$ soit

$$C_{k-1} = \sum_o^n c_i\ X^i \text{ et } A_{k-1,j} = \sum_o^n a_{ij}\ X^i,$$

nous aurons alors

$$C_k = -x_{k-1}c_o + \sum_1^n c_{i-1}x_{k-1}\ X^i$$

$$A_{k,j} = \sum_o^n (a_{ij} + Q(x_o,\dots,x_k;y_o,\dots,y_j)c_i)\ X^i.$$

A la fin de cette première passe, les polynômes $A_j(X)$ sont exprimés dans la base canonique. La deuxième passe consistera alors à exprimer les polynômes $D_j(Y)$ et $A_j(X)D_j(Y)$ dans la base canonique des X^iY^j. Pour cela, nous nous occuperons successivement des polynômes

$$B_o(X,Y) = A_o(X)$$

$$B_1(X,Y) = B_o(X,Y) + A_1(X)D_1(Y)$$

$$\vdots$$

$$B_m(X,Y) = B_{m-1}(X,Y) + A_m(X)D_m(Y).$$

Supposons déterminés les polynômes D_{k-1} et B_{k-1}, soit

$$D_{k-1} = \sum_o^m d_j\ Y^j.$$

Nous aurons alors

$$D_k = -y_{k-1}d_o + \sum_1^m (d_{j-1}-d_jy_{k-1})\ Y^j.$$

Pour déterminer le polynôme B_k, il nous suffira donc d'ajouter au polynôme B_{k-1} les polynômes

$-A_k y_{k-1} d_o$, $A_k(d_o - d_1)\ Y$, ..., $A_k(d_{m-1} - d_m y_{m-1})\ Y^m$.

Algorithme pour développer un polynôme à deux indéterminées

Cet algorithme nécessite $2 + 2\left[(n+1)+(m+1)+(n+1)(M+1)\right]$ mémoires qui sont n, m,

$x_o, \ldots, x_n$, $y_o, \ldots, y_m$, $c_o, \ldots, c_n$, $d_o, \ldots, d_m$, $z_{i,j}$, a_{ij} $\quad 0 \leqslant i \leqslant n$

$0 \leqslant j \leqslant m$.

0) (initialisation) données : $n \geqslant 1$, $m \geqslant 1$, $x_o, \ldots, x_n$ à deux distincts

$y_o, \ldots, y_m$ à deux distincts.

1) (calcul et mise en mémoire des $Q(x_i, y_j)$)

faire $Z_{i,j} = Q(x_i, y_j)$ $\quad 0 \leqslant i \leqslant n, \quad 0 \leqslant j \leqslant m.$

2) (calcul des différences divisées des $z_{i,j}$)

(première passe : à la fin de cette passe, on a $z_{i,j} = Q(x_o, \ldots, x_i; y_j)$)

```
pour j = 0,..., m faire
|  pour i = 1,...,n faire
|  |  pour k = i,...,n faire
|  |  |  z(k,j) = (z(k,j) - z(i-1,j)) / (x(k) - x(i-1))
|  |  refaire
|  refaire
refaire
```

(deuxième passe : à la fin de cette passe, on a $z_{i,j} = Q(x_o, \ldots, x_i; y_o, \ldots, y_j)$)

```
pour i = 0,...,n faire
|  pour j = 1,...,m faire
|  |  pour k = j,...,m faire
|  |  |  z(i,k) = (z(i,k) - z(i,j-1)) / (y(k) - y(j-1))
|  |  refaire
|  refaire
refaire
```

3) (on connaît maintenant le polynôme Q dans la base des $C_i D_j$)

(première passe)

(initialisation : $C_o = 1$, $A_{o,j} = Q(x_o; y_o, \ldots, y_j) C_o$)

faire $c_o = 1$, $c_1 = 0, \ldots, c_n = 0$

$a_{o,o} = z_{o,o}$, $a_{o,1} = z_{o,1}, \ldots, a_{o,m} = z_{o,m}$

$a_{i,j} = 0 \quad 1 \leqslant i \leqslant n \quad 0 \leqslant j \leqslant m$

(on exprime les polynômes C_i et $A_{i,j}$ dans la base des X^i)

pour $i = 1,\dots,n$ faire

  pour $k = n,\dots,1$ (dans cet ordre!) faire

    $c_k = c_{k-1} - x_{i-1}c_k$

    pour $j = 0,\dots,m$ faire

      $a_{k,j} = a_{k,j} + z_{i,j}c_k$

    refaire

  refaire

  faire $c_o = -x_{i-1}c_o$

  pour $j = 0,\dots,m$ faire

    $a_{o,j} = a_{o,j} + z_{i,j}c_o$

  refaire

refaire

(deuxième passe)

(initialisation : $D_o = 1$)

faire $d_o = 1,\ d_1 = 0,\dots,d_m = 0$

(on exprime les polynômes D_j dans la base des Y^j)

pour $j = 1,\dots,m$ faire

  pour $k = m,\dots,1$ (dans cet ordre!) faire

    $d_k = d_{k-1} - y_{j-1}d_k$

  refaire

  faire $d_o = -y_{j-1}d_o$

  pour $k = 0,\dots, j-1$ faire

    pour $i = 0,\dots,n$ faire

      $a_{i,k} = a_{i,k} + a_{i,j}\ d_k$

    refaire

  refaire

refaire

4) le polynôme cherché est le polynôme $Q = \sum_{i,j} a_{i,j}X^iY^j$.

6. QUELQUES REMARQUES POUR TERMINER

Voici quelques remarques qui peuvent être utiles lors de l'implémentation des algorithmes qui ont été décrits ici.

1) Supposons le polynôme $P(X)$ de la partie 2 à coefficients entiers. Si nous prenons soin de choisir pour $x_o,\dots,x_n$ des entiers, il est immédiat que les polynômes $P(x_o,X)$, $P(x_o,x_1,X),\dots$ sont eux aussi à coefficients entiers. Par conséquent, les différences divisées sont elles aussi des entiers. Si, au cours de ses calculs, notre calculatrice ne rencontre pas d'entier trop grand

qui l'oblige à passer en notation "scientifique" (et par là à perdre les derniers chiffres de cet entier), le résultat affiché sera exact. Même conclusion en ce qui concerne le polynôme $Q(X,Y)$ de la partie 5 (pourvu, évidemment, que $y_o,\ldots,y_m$ soient des entiers).

2) Pour que notre calculatrice manipule les entiers les plus petits possibles, il est bon de choisir

$$x_o = -[n/2] \;,\; x_1 = -[n/2]+1,\ldots,x_n = -[n/2]+n$$
$$y_o = -[m/2] \;,\; y_1 = -[m/2]+1,\ldots,y_m = -[m/2]+m.$$

Avec un SHARP PC 1211 ou 1500, on arrive ainsi à développer un polynôme à une indéterminée de degré 7 à 8, pourvu que les coefficients soient de taille raisonnable. Par contre, le polynôme $P(X) = X^{12}$ donne à l'affichage $X^{12} + 2X$! (la publicité est gratuite). Pour se prémunir contre de tels résultats fantaisistes, il est bon de placer partout où cela est nécessaire des détecteurs d'overflow dans le programme.

3) Une astuce pour développer un polynôme à une indéterminée et de degré élevé. Supposons que je veuille développer le polynôme

$$P(X) = (X^3 - 3X^2 + 6X + 5)^4(2X^3 + 8X^2 + X - 15)$$

Je peux tout d'abord développer le polynôme

$$Q(X,Y) = (X^2Y - 3XY + 6X + 5)^2(XY^2 - 3XY + 6Y + 5)^2(2X^2Y + 8XY + X - 15)$$

l'idée étant de dédoubler les monômes et X de manière à obtenir un polynôme $Q(X,Y)$ dont les degrés en X et Y sont raisonnables. Une fois le polynôme $Q(X,Y)$ développé, il ne reste plus qu'à utiliser l'identité $P(X) = Q(X,X)$ pour obtenir le développement du polynôme $P(X)$. Toujours sur SHARP PC 1500 on arrive ainsi à développer des polynômes dont le degré va jusqu'à 14. Le prix à payer : il faut un module mémoire supplémentaire, car le programme correspondant nécessite entre 2K à 3K de mémoire.

Pour ceux que cela intéresse, voici les développements des divers polynômes :

- partie 0 :

$$P(X) = 84X^6 - 53X^5 - 549X^4 - 48\,316X^3 - 1\,489\,319X^2 - 23\,088\,267X - 143\,145\,627$$

- partie 5 :

$$Q(X,Y) = -X^6 + 35X^4 + 6X^3 - (8X^5 - 4X^4 - 9X^3 - 45X^2)Y - (39X^3 + 31X^2 - 107X + 3)Y^2 - (X^2 + 292X - 161)Y^3 + (99X - 221)Y^4 + 68Y^5$$

- partie 6 :

$$P(X) = 2X^{15} - 16X^{14} + 61X^{13} - 11X^{12} - 644X^{11} - 2\,558X^{10} - 2\,511X^9 - 6\,639X^8 + 27\,114X^7 - 19\,394X^6 - 27\,494X^5 + 67\,130X^4 + 37\,850X^3 - 50\,500X^2 - 44\,375X - 9\,375$$