

Algorithmes pour la sommation et l'intégration symboliques

Lucien Pech, sous la direction de Frédéric Chyzak,
Laboratoire joint INRIA – Microsoft Research

23 août 2009 (mise à jour du 13 novembre 2009)

Le but du stage était d'implanter l'algorithme de Chyzak [9] (2000), ainsi qu'une part de son extension à une classe plus large de fonctions par Chyzak, Kauers et Salvy [10] (2009) en Maple. Cette implantation devait en outre permettre d'étudier et de comparer pratiquement les différentes variantes algorithmiques proposées depuis dix ans [4, 5, 7].

Notations

Nous noterons toujours en lettres grasses des n -uplets (ou des séquences suivant le contexte); par exemple $\boldsymbol{\partial} = (\partial_1, \dots, \partial_n)$, $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$... La notation $\boldsymbol{\partial}^\alpha$ désigne le produit $\partial_1^{\alpha_1} \dots \partial_n^{\alpha_n}$. On notera $|\boldsymbol{\alpha}| = |\alpha_1| + \dots + |\alpha_n|$.

Introduction

Fonctions spéciales, fonctions ∂ -finies, systèmes fonctionnels linéaires

On sait par le théorème de Cauchy que, sous certaines conditions de régularité, une équation différentielle munie d'un nombre convenable de conditions initiales admet une unique solution. L'équation différentielle permet donc de représenter implicitement sa solution.

L'algorithmique développée ici traite des solutions d'équations différentielles à coefficients polynomiaux. Ce cadre peut sembler très restrictif, mais dans la pratique, la plupart des fonctions spéciales "usuelles" rentrent dans ce moule : c'est par exemple le cas de plus de 60% des fonctions de [3], et notamment des fonctions algébriques, de l'exponentielle, des fonctions trigonométriques, des polynômes orthogonaux, des fonctions de Bessel...

Dans le cas des suites, le rôle des équations différentielles est joué par des récurrences à coefficients polynomiaux.

Algèbres de Ore

Les traitements des équations différentielles, des récurrences et des q -récurrences sont très similaires. Le bon formalisme pour traiter ces trois cas de façon uniforme est l'utilisation des algèbres de Ore. Elles ont d'abord été introduites par Ore [17], et définies de la façon qui suit par Chyzak et Salvy [11]. Elles ont été abordées dans le cours 2.22 (*cf.* chapitre 22 de [6]); nous nous cantonnerons donc aux définitions.

Les algèbres de Ore sont des structures algébriques dont les éléments sont des opérateurs, représentant des équations différentielles, des récurrences...

Définition 1 (Algèbre de Ore rationnelle ; cf. [6] p. 213). Soit k un corps de caractéristique nulle. Étant donnés :

- le corps $k(\mathbf{x}) = k(x_1, \dots, x_n)$ de fractions rationnelles en x_1, \dots, x_n ,
- n endomorphismes σ_i de ce corps commutant deux à deux,
- pour chaque i une σ -dérivation δ_i relative à σ_i , c'est-à-dire pour chaque i un endomorphisme linéaire tel que $\forall a, b \in k(\mathbf{x}), \delta_i(ab) = \sigma_i(a)\delta_i(b) + \delta_i(a)b$, toutes ces σ -dérivations commutant deux à deux et δ_i commutant avec σ_j pour tout $i \neq j$,
- n indéterminées ∂_i ,

l'algèbre de Ore rationnelle notée $k(x_1, \dots, x_n)\langle \partial_1, \dots, \partial_n; \sigma_1, \dots, \sigma_n, \delta_1, \dots, \delta_n \rangle$ ou plus simplement $k(\mathbf{x})\langle \partial \rangle$ est la $k(\mathbf{x})$ -algèbre associative engendrée par les ∂_i modulo les relations :

$$\forall a \in k(\mathbf{x}), \partial_i a = \sigma_i(a)\partial_i + \delta_i(a), \quad \partial_i \partial_j = \partial_j \partial_i.$$

EXEMPLE. On note I l'identité, D_x la dérivation par rapport à x , S_n l'opérateur de décalage par rapport à la variable n (défini par $S_n(u)(n) = u(n+1)$), $\Delta_n = S_n - I$ l'opérateur de différence finie, $H_{q,x}$ l'opérateur de q -dilatation par rapport à la variable x (défini par $H_{q,x}(u)(x) = u(qx)$), $\Delta_{q,x} = H_{q,x} - I$ l'opérateur de q -différence finie. On utilise les algèbres de Ore rationnelles suivantes :

- Différentiel : $k(x)\langle \partial; I, D_x \rangle$. On a $\partial x = x\partial + 1$.
- Différence finie : $k(n)\langle \partial; S_n, \Delta_n \rangle$. On a $\partial n = (n+1)\partial + 1$.
- q -différence finie : $k(q)(x)\langle \partial; H_{q,x}, \Delta_{q,x} \rangle$. On a $\partial x = qx\partial + (q-1)x$.

Les éléments d'une algèbre de Ore A représentent donc des opérateurs, dont on définit une action sur un ensemble de fonctions, de suites, de séries, etc., qui constitue alors un A -module à gauche.

EXEMPLE.

- $A = k(x)\langle \partial; I, D_x \rangle$. $\partial \cdot f(x) = \frac{df}{dx}(x)$, $x \cdot f(x) = xf(x)$.
- $A = k(n)\langle \partial; S_n, \Delta_n \rangle$. $\partial \cdot f(n) = f(n+1) - f(n)$, $n \cdot f(n) = nf(n)$.
- $A = k(q)(x)\langle \partial; H_{q,x}, \Delta_{q,x} \rangle$. $\partial \cdot f(x) = f(qx) - f(x)$, $x \cdot f(x) = xf(x)$.

Fonctions ∂ -finies

Dans cette partie, on fixe une algèbre de Ore rationnelle $A = k(\mathbf{x})\langle \partial \rangle$.

L'ensemble des opérateurs de A qui annulent une fonction f est un idéal à gauche de A noté $\text{ann}(f)$: par exemple dans le cas différentiel, multiplier par x ou dériver la fonction nulle donne la fonction nulle.

La plupart des algorithmes présentés dans ce rapport traitent des fonctions ∂ -finies, qui intuitivement sont les fonctions définies par un ensemble fini d'équations et de conditions initiales :

Définition 2 (∂ -finie). Une fonction $f(\mathbf{x})$ est ∂ -finie sous l'action de A si $A \cdot f$ est un $k(\mathbf{x})$ -espace vectoriel de dimension finie. Dans ce cas, $A/\text{ann}(f)$ est un espace vectoriel de dimension finie.

Les fonctions ∂ -finies ont de bonnes propriétés de clôture :

Proposition 1. Soient f, g deux fonctions ∂ -finies sous l'action de A . Alors $f + g$, fg , ∂f sont ∂ -finies sous l'action de A .

Cette propriété est constructive! Étant donné un ensemble fini de générateurs des idéaux annulateurs de deux fonctions f et g , on dispose d'algorithmes effectifs pour calculer un générateur des idéaux annulateurs de $f + g$, fg , ∂f [11]. Ces algorithmes ont été implantés par Frédéric Chyzak dans Mgfun. Mon code s'appuie sur une fonction de Mgfun qui prend en entrée une fonction ∂ -finie et retourne un générateur de l'idéal annulateur.

Une sous-classe des fonctions ∂ -finies que nous ne détaillerons pas dans ce rapport est celle des *fonctions holonomes*. Les algèbres de Ore considérées dans ce cas sont à coefficients dans $k[\mathbf{x}]$ au lieu de $k(\mathbf{x})$. On définit de même l'idéal à gauche annulateur de $f : \text{ann}(f)$, et on dit que f est holonome sous l'action de A si le nombre de fonctions linéairement indépendantes parmi les $\mathbf{x}^{\mathbf{m}}\partial^{\mathbf{k}}f$ pour $|\mathbf{m}| + |\mathbf{k}| \leq N$ croît en $O(N^n)$. Les preuves de terminaison de l'algorithme de création télescopique de Chyzak (2000) ne s'appliquent qu'aux fonctions holonomes.

Généralisation aux fonctions non ∂ -finies

L'algorithme de Chyzak que nous détaillerons dans la partie 1.2 s'applique aux fonctions ∂ -finies. Chyzak, Kauers et Salvy ont montré dans [10] que le principe sous-jacent de l'algorithme s'applique à une classe plus large de fonctions, en s'appuyant sur la notion de dimension de Hilbert d'un idéal :

Définition-proposition 1 (dimension de Hilbert). Soit $A = k(\mathbf{x})\langle\partial\rangle$ algèbre de Ore rationnelle. On note $A^{(d)}$ l'ensemble de polynômes de A de degré total en ∂ inférieur à d . Soient I un idéal de A , $I^{(d)} = I \cap A^{(d)}$. Soit $\text{HF}_I(d) = \dim(A^{(d)}/I^{(d)}) = \dim(A^{(d)}) - \dim(I^{(d)})$. Pour d assez grand, $\text{HF}_I(d)$ est un polynôme en d , dont le degré est appelé *dimension de Hilbert* de l'idéal, noté $\dim(I)$.

EXEMPLE. L'idéal annulateur d'une fonction ∂ -finie est de dimension de Hilbert nulle.

EXEMPLE. On note $f(n, k) = \left\{ \begin{smallmatrix} n \\ k \end{smallmatrix} \right\}$ les nombres de Stirling de seconde espèce, définis comme le nombre de partitions d'un ensemble de cardinal n en k ensembles non vides. L'idéal $\text{ann}(f)$ est généré par $S_n S_k - 1 - (k+1)S_k$ dans $\mathbb{Q}(n, k)\langle S_n, S_k \rangle$. f n'est pas ∂ -finie, $\dim(\text{ann}(f)) = 1$.

On peut étendre la propriété 1 de la façon suivante :

Proposition 2 ([10], THEOREM 1). Soient I_1, I_2 deux idéaux à gauche de A , f_1, f_2 des fonctions annulées par I_1 et I_2 , respectivement. Alors :

1. $\dim(\text{ann}(\partial \cdot f_1)) \leq \dim(I_1)$,
2. $\dim(\text{ann}(f_1 + f_2)) \leq \max(\dim(I_1), \dim(I_2))$,
3. $\dim(\text{ann}(f_1 f_2)) \leq \dim(I_1) + \dim(I_2)$.

Cette propriété est également constructive. Des algorithmes sont donnés dans [10], mais qui ne calculent que des sous-idéaux de $\text{ann}(\partial \cdot f_1)$, $\text{ann}(f_1 + f_2)$ et $\text{ann}(f_1 f_2)$.

1 Algorithme de création télescopique

1.1 Principe de la création télescopique

On présente le principe de la création télescopique dans le cas des suites.

1.1.1 Un premier exemple

On veut calculer $u_n = \sum_k \binom{n}{k}$. De la relation de Pascal $\binom{n+1}{k+1} = \binom{n}{k+1} + \binom{n}{k}$, on déduit que $\binom{n+1}{k+1} = 2\binom{n}{k} + \binom{n}{k+1} - \binom{n}{k}$. En sommant cette dernière égalité pour tous les k entiers ($\binom{n}{k} = 0$ pour $k < 0$ et $k > n$), les termes $\binom{n}{k+1} - \binom{n}{k}$ se télescopent, et on obtient $u_{n+1} = 2u_n$. Avec $u_0 = 1$, on en déduit $u_n = 2^n$.

1.1.2 Généralisation

On considère une suite $u_{n,k}$ à deux indices. On pose $U_n = \sum_{k=a}^b u_{n,k}$, et on cherche une récurrence vérifiée par U_n . Pour cela, on cherche des opérateurs $P(n, S_n)$ et $Q(n, k, S_n, S_k)$ tels que $Pu = \Delta_k Q(u)$.

On somme cette dernière égalité pour tous les k entiers compris entre a et b . Le signe \sum et l'opérateur P commutent car P ne dépend pas de k , et on obtient $PU_n = Q(u)(b+1) - Q(u)(a)$.

1.2 Algorithme détaillé

Entrée

On se place dans l'algèbre de Ore rationnelle $A = \mathbb{Q}(\mathbf{x})\langle \partial \rangle$. Soit $f(\mathbf{x})$ une fonction ∂ -finie sous l'action de A , et soit V un $\mathbb{Q}(\mathbf{x})$ -espace vectoriel de dimension finie, stable sous l'action de $\partial_1, \dots, \partial_n$, de base $B = (b_1, \dots, b_r)$, contenant f (généralement on prendra $V = \text{Vect}(\partial^\alpha f, \alpha \in \mathbb{N}^n)$, mais il sera parfois nécessaire de considérer un espace vectoriel plus grand).

L'algorithme prend en entrée les coordonnées de f dans la base B , les matrices des actions de $\partial_1, \dots, \partial_n$ dans cette base (où la matrice de l'action de ∂_i est la matrice $[\partial_i b_1 | \dots | \partial_i b_r]$, avec $\partial_i b_j$ vecteur exprimé dans la base B), et un entier positif D : la dimension de Hilbert maximale de l'idéal engendré par les opérateurs P retournés par l'algorithme.

EXEMPLE. On prend $f(n, x) = \frac{x \exp(x) T_n(x)}{\sqrt{1-x^2}}$ (avec T_n le n -ème polynôme de Tchebycheff, défini par $\cos(nx) = T_n(\cos(x))$). L'algèbre considérée est $A = \mathbb{Q}(n, x)\langle \Delta_n, D_x \rangle$. f est annihilée par les opérateurs $\Delta_n^2 + 2(1-x)\Delta_n + 2(1-x)$ et $D_x - G(n, x) + \frac{n}{1-x^2}\Delta_n$, avec $G(n, x) = \frac{-x^3 + nx^2 - (n-1)x + 1}{x(1-x^2)}$. L'espace vectoriel V de base $B = (f, \Delta_n f)$ est donc stable sous l'action de Δ_n et D_x . Les matrices des actions de D_x et Δ_n sur V sont donc :

$$\begin{bmatrix} G(n, x) & 2\frac{n+1}{x+1} + G(n+1, x) - G(n, x) \\ -\frac{n}{1-x^2} & G(n+1, x) - \frac{1}{1-x^2} + 2\frac{n+1}{x+1} \end{bmatrix} \text{ et } \begin{bmatrix} 0 & -2(1-x) \\ 1 & -2(1-x) \end{bmatrix}.$$

Sortie

L'algorithme retourne une famille d'opérateurs $P_i \in \mathbb{Q}(x_1, \dots, x_{n-1})[\partial_1, \dots, \partial_{n-1}]$ et de vecteurs $q_i \in \mathbb{Q}(\mathbf{x})^r$ tels que pour tout i on ait $P_i f = \partial_n(\sum_{j=1}^r q_{ij} b_j)$, et que la dimension de Hilbert de l'idéal de A engendré par les P_i soit au plus égale à D .

EXEMPLE. Pour la fonction f de l'exemple précédent, l'algorithme retourne :

$$P = -n(n^2 + 3n + 2) + \Delta_n(-4n - n^3 - 4n^2 - 2) - 1/2\Delta_n^2(1 + n^2 + n),$$

$$q = \left(-\frac{n+2-2nx-3x-xn^2+x^2+x^2n^2+nx^2}{x}, -\frac{n+1-x+x^2+x^2n^2+nx^2}{x} \right).$$

Détails

Soit d un entier strictement positif. On détaille la recherche d'un opérateur P de degré total d en les ∂_i .

Soient $I_d = \{\alpha \in \mathbb{N}^{n-1} : |\alpha| \leq d\}$, $N_d = |I_d| = \binom{d+n-2}{n-1}$. On introduit N_d coefficients inconnus $\eta_\alpha \in \mathbb{Q}(x_1, \dots, x_{n-1})$ avec $\alpha \in I_d$, ainsi que r inconnues $\phi_i \in \mathbb{Q}(\mathbf{x})$ pour $i \in \llbracket 1, r \rrbracket$. On pose $P = \sum_{\alpha \in I_d} \eta_\alpha \partial^\alpha$ et $q = (\phi_1, \dots, \phi_r)$.

La première étape de l'algorithme consiste à exprimer Pf dans la base B à l'aide des matrices des actions des ∂_i .

EXEMPLE. On prend $d = 2$ dans l'exemple précédent. On a $Pf = \eta_0(n)f + \eta_1(n)\Delta_n f + \eta_2(n)\Delta_n^2 f = (\eta_0(n) - 2(1-x)\eta_2(n))f + (\eta_1(n) - 2(1-x)\eta_2)\Delta_n f$.

On pose ensuite $g(\mathbf{x}) = \sum_{i=1}^r \phi_i(\mathbf{x})b_i(\mathbf{x})$, et on exprime $\partial_n g$ dans la base B , toujours avec les matrices des actions des ∂_i . On identifie alors cette dernière expression à celle de Pf , coordonnée par coordonnée, ce qui fournit un système d'équation.

Il est alors nécessaire de distinguer les cas suivant la nature de l'opérateur ∂_n , pour se ramener à un système de la forme $\partial_n \Phi = A\Phi + B_d E_d$, avec $A \in M_r(\mathbb{Q}(\mathbf{x}))$ et $B \in M_{r \times N_d}(\mathbb{Q}(\mathbf{x}))$, et d'inconnues $\Phi = (\phi_1, \dots, \phi_r)$ et $E_d = (\eta_\alpha)_{\alpha \in I_d}$.

Prenons l'exemple du cas différentiel, avec $\partial_n = D_{x_n}$. Soit M la matrice de l'action de ∂_n dans la base B . $\partial_n g = \sum_{i=1}^r \frac{\partial \phi_i}{\partial x_n}(\mathbf{x})b_i(\mathbf{x}) + \phi_i(\mathbf{x})\frac{\partial b_i}{\partial x_n}(\mathbf{x})$. Donc $\partial_n \Phi = -M\Phi + Pf$.

EXEMPLE. $D_x \Phi = - \begin{bmatrix} G(n, x) & 2\frac{n+1}{x+1} + G(n+1, x) - G(n, x) \\ -\frac{n}{1-x^2} & G(n+1, x) - \frac{1}{1-x^2} + 2\frac{n+1}{x+1} \end{bmatrix} \Phi + \begin{bmatrix} 1 & 0 & -2(1-x) \\ 0 & 1 & -2(1-x) \end{bmatrix} E_2$, avec $E_2 = {}^t(\eta_0, \eta_1, \eta_2)$.

L'espace des solutions (Φ, E_d) du système est un espace vectoriel. On note $(P_i, q_i)_i$ une base de cet espace.

Choix de la valeur de d

On calcule, pour d croissant, une famille $(P_i, q_i)_i$ comme à la section précédente, jusqu'à ce que l'idéal engendré par les P_i dans A soit de dimension de Hilbert inférieure à D .

Dans le cas où f est non seulement ∂ -finie mais en plus holonome, on peut prouver par des arguments combinatoires qu'il existe une valeur de d pour laquelle l'idéal engendré par les P_i est de dimension nulle.

J'ai également expérimenté l'approche consistant à doubler la valeur de d à chaque étape ; mais le coût du calcul augmentant rapidement avec la valeur de d , dans le plupart des exemples la perte liée au calcul d'un opérateur d'ordre trop grand n'était pas compensée par le calcul de moins d'opérateurs.

1.3 Extension au cas non ∂ -fini

Lorsque $\text{ann}(f)$ est de dimension de Hilbert strictement positive, $A/\text{ann}(f)$ est de dimension vectorielle infinie. On ne peut donc pas trouver d'espace vectoriel V contenant f stable sous l'action de $\partial_1, \dots, \partial_n$ (dont un sous-espace est isomorphe à $A/\text{ann}(f)$).

Commençons par reformuler l'algorithme de la section précédente en termes d'opérateurs et d'idéaux. On pose une forme *a priori* de l'opérateur Q comme combinaison linéaire des monômes irréductibles par $\text{ann}(f)$, et de P comme combinaison linéaire des monômes en $\partial_1, \dots, \partial_{n-1}$ de degré inférieur à d . On réduit $T = P - \partial Q$ modulo $\text{ann}(f)$ (*i. e.* on calcule les composantes de $T \bmod \text{ann}(f)$ dans une base de l'espace vectoriel de dimension finie $A/\text{ann}(f)$). On veut que $T \bmod \text{ann}(f) = 0$, donc $T \bmod \text{ann}(f)$ doit être nul composante par composante dans cette base, ce qui fournit le système d'équations couplé décrit précédemment.

Dans le cas où $A/\text{ann}(f)$ est de dimension vectorielle infinie, il y a une infinité de monômes irréductibles par $\text{ann}(f)$. On restreint la forme *a priori* de Q aux termes de degré inférieur à d . On réduit de même $T = P - \partial Q$ modulo $\text{ann}(f)$ sur une base de $A/\text{ann}(f)$, les coordonnées de T sont alors presque toutes nulles. Celles qui restent fournissent un système, qui contrairement au cas précédent est surdéterminé (il y a plus d'équations de d'inconnues).

Si P, Q sont tels que $P - \partial Q = 0$ modulo $\text{ann}(f)$, et que $\max(\deg(P), \deg(Q)) \leq d$, alors l'algorithme précédent trouve P et Q . Un résultat théorique, dont l'argument essentiel est la proposition 2, permet de prédire la dimension de Hilbert de l'idéal engendré par les P ([10] THEOREM 3). On applique donc l'algorithme pour des valeurs de d croissantes jusqu'à ce que les P calculés engendrent un idéal dont la dimension de Hilbert est égale à celle prédite.

1.4 Exemples d'applications

1.4.1 Application au calcul concret d'une intégrale

Soit $f(x, a) = xJ_1(ax)I_1(ax)Y_0(x)K_0(x)$. Dans [15], Glasser et Montaldi donnent sans preuve l'égalité :

$$\int_0^\infty f(x, a)dx = -\frac{\ln(1-a^4)}{2\pi a^2}, \quad 0 < a < 1.$$

f est ∂ -finie : les dérivées de f par rapport à x et a sont combinaisons linéaires des $\frac{\partial^{i+j}}{\partial x^i \partial a^j} f(x, a)$, pour $i, j \leq 3$. On peut donc appliquer notre fonction `creative_telescoping` :

```
> f := x*BesselJ(1,a*x)*BesselI(1,a*x)*BesselY(0,x)*BesselK(0,x):
> ct := Mgfun:-creative_telescoping(f, a::diff, x::diff);
```

$$\begin{aligned} ct := & \left[\left[\frac{8(a^4-1)}{a} F(a) + 4(a^4-1) \frac{d}{da} F(a), \frac{2(-1+8a^4x^4)}{ax^3} f(x, a) + \frac{2}{ax^2} \frac{\partial}{\partial x} f(x, a) \right. \right. \\ & - \frac{19}{x^3} \frac{\partial}{\partial a} f(x, a) - \frac{4a}{x} \frac{\partial^4}{\partial a^2 \partial x^2} f(x, a) - \frac{70a}{x^3} \frac{\partial^2}{\partial a^2} f(x, a) + \frac{2}{a} \frac{\partial^3}{\partial x^3} f(x, a) - \frac{16a^2}{x^3} \frac{\partial^3}{\partial a^3} f(x, a) \\ & \left. \left. + \frac{25}{x^2} \frac{\partial^2}{\partial a \partial x} f(x, a) + \frac{32a}{x^2} \frac{\partial^3}{\partial a^2 \partial x} f(x, a) - \frac{12}{x} \frac{\partial^3}{\partial a \partial x^2} f(x, a) + \frac{\partial^4}{\partial a \partial x^3} f(x, a) + \frac{6a^2}{x^2} \frac{\partial^4}{\partial a^3 \partial x} f(x, a) \right] \right] \end{aligned}$$

On a donc $P(f) = D_x Q(f)$ avec $P = 4(a^4-1)D_a + \frac{8(a^4-1)}{a}$ et $Q = \frac{2(-1+8a^4x^4)}{ax^3} + \frac{2}{ax^2} D_x - \frac{19}{x^3} D_a - \frac{4a}{x} D_a^2 D_x^2 - \frac{70a}{x^3} D_a^2 + \frac{2}{a} D_x^3 - \frac{16a^2}{x^3} D_a^3 + \frac{25}{x^2} D_a D_x + \frac{32a}{x^2} D_a^2 D_x - \frac{12}{x} D_a D_x^2 + D_a D_x^3 + \frac{6a^2}{x^2} D_a^3 D_x$. Si on pose $g(x, a) = Q(f)$ et $F(a) = \int_0^\infty f(x, a)dx$, en intégrant la relation précédente par

rapport à x et en intervertissant l'opérateur P (qui ne dépend pas de x !) et le signe intégrale on a donc $PF = [g(x, a)]_{x=0}^{x=\infty}$.

On calcule la limite de g à l'aide du package `MultiSeries` :

```
> g := normal(eval(ct[1][2], _f(x,a)=f)):
> MultiSeries[limit](g, x=infinity) assuming 0<a and a<1;
0
> MultiSeries[limit](g, x=0, right);
8a
π
```

On a donc $F'(a) = -\frac{2}{a}F(a) + \frac{2a}{\pi(1-a^4)}$, qui admet pour solutions les $F(a) = \frac{C}{a^2} - \frac{\ln(1-a^4)}{2\pi a^2}$ pour $0 < a < 1$, avec C constante fixée.

Pour $x > 0$ fixé $f(x, a) \xrightarrow{a \rightarrow 0} 0$, donc par convergence dominée $F(a) \xrightarrow{a \rightarrow 0} 0$, donc $C = 0$ et $F(a) = -\frac{\ln(1-a^4)}{2\pi a^2}$.

1.4.2 Création télescopique itérée

L'entrée de l'algorithme est une famille d'opérateurs qui annulent une fonction, et la sortie est une famille d'opérateurs qui annulent la somme (resp. l'intégrale) de la fonction.

On peut donc itérer le procédé, et sommer ou intégrer successivement par rapport à différentes variables. J'ai implanté une fonction `iterated_creative_telescoping` qui effectue automatiquement cette tâche lorsque le second membre $Q(f)$ est nul aux extrémités de l'intervalle considéré. Par exemple, on peut montrer que :

$$\sum_{r=0}^{\infty} \sum_{s=0}^{\infty} (-1)^{(n+r+s)} \binom{n}{r} \binom{n}{s} \binom{n+r}{r} \binom{n+s}{s} \binom{2n-r-s}{n} = \sum_{k=0}^{\infty} \binom{n}{k}^4. \quad (1)$$

On somme d'abord $(-1)^{(n+r+s)} \binom{n}{r} \binom{n}{s} \binom{n+r}{r} \binom{n+s}{s} \binom{2n-r-s}{n}$ par rapport à s , puis par rapport à r . Cette expression est nulle pour s ou r strictement négatif ou supérieur à n , on peut donc appliquer notre fonction `iterated_creative_telescoping` :

```
> Mgfun:-iterated_creative_telescoping((-1)^(n+r+s)*binomial(n,r)
*binomial(n,s)*binomial(n+r,r)*binomial(n+s,s)*binomial(2*n-r-s,n),
n::shift, [s::shift, r::shift])[1][1];
-4 * (16*n^3 + 48*n^2 + 47*n + 15) * _F(n) / (n^3 + 6*n^2 + 12*n + 8)
- 2 * (6*n^3 + 27*n^2 + 41*n + 21) * _F(n+1) / (n^3 + 6*n^2 + 12*n + 8)
+ _F(n+2)
```

Pour vérifier l'identité (1), on s'assure ensuite que $\binom{n}{k}^4$ est bien annulé par le même opérateur :

```
> Mgfun:-creative_telescoping(binomial(n,k)^4, n::shift, k::shift)[1][1];
(-4*n - 3) * _F(n) - 1/2 * (6*n^3 + 27*n^2 + 41*n + 21) * _F(n+1) / (4*n^2 + 9*n + 5)
+ 1/4 * (n^3 + 6*n^2 + 12*n + 8) * _F(n+2) / (4*n^2 + 9*n + 5)
```

On retrouve bien le même opérateur à un facteur multiplicatif près. Il ne reste donc plus qu'à vérifier que les membres de gauche et de droite de l'identité (1) sont bien égaux pour $n = 0, 1$, ce qui se vérifie aisément.

2 Résolution rationnelle de systèmes

La partie centrale de l'algorithme de création télescopique est la résolution d'une famille de systèmes de la forme $\partial y = Ay + B_d e_d$, où ∂ est un opérateur différentiel, de différence finie ou de q -différence finie en la variable t , d'inconnues $y \in \mathbb{Q}(x, t)^n$ et $e_d \in \mathbb{Q}(x)^d$, avec $A \in M_n(\mathbb{Q}(x, t))$ indépendant de d et $B \in M_{n \times d}(\mathbb{Q}(x, t))$.

Il n'y a pas à l'heure actuelle de résultat théorique sur le coût de cette opération. Notre approche a été exclusivement expérimentale.

2.1 Cas scalaire

Le cas le plus simple est celui où la dimension $n = 1$. Ce cas est important, car les techniques de découplage (section 2.2) ramène la résolution du système à la résolution d'équations scalaires, et l'approche directe (section 2.3) est une généralisation de l'approche scalaire.

Les cas différentiel et de différence finie ont été traités dans le cours 2.22 du MPRI (*cf.* chapitres 18 et 19 du polycopié [6]).

Dans chacun des trois cas (différentiel, différence finie et q -différence finie), le calcul se décompose en trois grandes étapes :

1. Calcul d'un multiple des dénominateurs des solutions. Un changement d'inconnue ramène alors la recherche du numérateur à celle des solutions polynomiales d'une nouvelle équation.
2. Calcul d'une borne sur le degré des solutions polynomiales.
3. Calcul de la solution par coefficients indéterminés (méthode classique, connue de Newton), ou bien en exploitant la récurrence entre les coefficients des solutions [1, 5, 7].

Dans ce qui suit nous détaillons ces trois étapes dans le cas de la q -différence finie. On notera $\mathbb{F} = \mathbb{Q}(q)$. Suivant la question posée, il sera parfois plus simple de considérer un opérateur en $H_{q,x}$, et d'autres fois un opérateur en $\Delta_{q,x}$; on passe aisément d'un opérateur à l'autre par $H_{q,x}^k = \sum_{i=0}^k \binom{k}{i} \Delta_{q,x}^i$ et $\Delta_{q,x}^k = \sum_{i=0}^k (-1)^{k-i} \binom{k}{i} H_{q,x}^i$.

2.1.1 Multiple du dénominateur

Soit $L = \sum_{i=0}^r a_i(x) H_{q,x}^i$, avec $a_i \in \mathbb{F}[x]$ pour $i \in \llbracket 0, r \rrbracket$. On cherche un multiple $D \in \mathbb{F}[x]$ de tous les dénominateurs $Q \in \mathbb{F}[x]$ tels qu'il existe un numérateur $P \in \mathbb{F}[x]$ premier avec Q tel que $L(P/Q)$ soit polynomial.

Soient $P, Q \in \mathbb{F}[x], k \in \mathbb{N}, u = P/(x^k Q)$, avec $x \nmid Q$ et P et $x^k Q$ premiers entre eux, et tels que Lu soit polynomial.

Borne sur k . Soit $v = \min_i(\text{val}(a_i))$ la valuation de L . Lu est de valuation supérieure à $v - k$, son coefficient de degré $v - k$ étant $\frac{P(0)}{Q(0)} \sum_{i=0}^r a_{i,v} q^{-ik}$ (où $a_{i,v}$ est le coefficient de degré v de a_i), qui doit être nul si $k > v$ (Lu doit être polynomial). Donc si $k > v$, en multipliant cette expression par q^{kr} on obtient $\sum_{i=0}^r a_{i,v} q^{(r-i)k} = 0$ (si $k > 0, P(0) \neq 0$); donc si on pose $p(X) = \sum_{i=0}^r a_{i,v} X^{r-i} \in \mathbb{F}[X]$, une borne sur k est donc donnée par $\max(v, \max(n \geq 0 : p(q^n) = 0))$.

EXEMPLE. Soit $L = -(qx-1) + q^4(q^5x-1)H_{q,x}^2$. On a $v = 0$, $p(X) = X^2 - q^4 = (X - q^2)(X + q^2)$, donc $k \leq 2$. En substituant u par u/x^2 dans L et en multipliant par x^2 pour obtenir des coefficients polynomiaux on est ramené à considérer l'opérateur $L' = -(qx-1) + (q^5x-1)H_{q,x}^2$.

Multiple de Q . On a maintenant $u = P/Q$, $x \nmid Q$. On observe que les pôles de $u, H_{q,x}u, \dots, H_{q,x}^r u$ sont décalés les uns par rapport aux autres d'un facteur q^n avec n entier strictement positif. Si u n'a pas deux pôles différant d'un facteur q^n , il ne peut y avoir de solution rationnelle que si $Q(x) \mid a_0(x), Q(qx) \mid a_1(x), \dots, Q(q^r x) \mid a_r(x)$, et dans ce cas un multiple du dénominateur est donné par $\text{pgcd}(a_0(x), a_1(qx), \dots, a_r(q^r x))$.

Dans le cas général, il peut y avoir des pôles de u qui diffèrent d'un facteur q^n . Soient $\alpha, \alpha q^N$ des racines de Q avec N maximal. $Lu = \sum_{i=0}^r a_i(x)P(xq^i)/Q(xq^i)$ est polynomial. En l'évaluant en $x = xq^N$, comme $Q(xq^{N+1}) \neq 0, \dots, Q(xq^{N+r}) \neq 0$ par maximalité de N , on a $a_0(xq^N) = 0$. De même, $a_r(\alpha) = 0$. On pose donc $R(w) = \text{Res}_x(a_0(wxq^r), a_r(x))$. Deux racines de Q diffèrent d'un facteur q^k , avec k ou plus égal à $K = \max(n \geq 0 : R(q^n) = 0)$.

EXEMPLE. $L' = -(qx-1) + (q^5x-1)H_{q,x}^2$. $R(w) = \text{Res}_x(-(q^3wx-1), q^5x-1) = q^3(w-q^2)$, donc $K = 2$.

On pose alors $M(x) = \text{ppcm}(Q(xq), Q(xq^2), \dots, Q(xq^r))$. En multipliant Lu par $M(x)$, on obtient que $\sum_{i=0}^r a_i(x)P(xq^i) \frac{M(x)}{Q(xq^i)}$ est un polynôme, donc $Q(x) \mid a_0(x)M(x)$ ($\frac{M(x)}{Q(xq^i)}$ est un polynôme pour $i \in \llbracket 1, r \rrbracket$). En évaluant cette dernière relation en $x = qx$, on obtient $Q(x) \mid a_0(x)a_1(x)\text{ppcm}(Q(xq^2), \dots, Q(xq^{r+1}))$. Par récurrence on a donc $Q(x) \mid a_0(x)a_1(x) \dots a_j(x)\text{ppcm}(Q(xq^{j+1}), \dots, Q(xq^{j+r}))$. Comme $\text{pgcd}(Q(x), Q(xq^j)) = 1$ pour $j > K$ par maximalité de N , on a donc $Q(x) \mid a_0(x)a_1(xq) \dots a_0(xq^K)$.

De même, en multipliant Lu par $\text{ppcm}(Q(x), Q(xq^{-1}), \dots, Q(xq^{-r+1}))$ on obtient $Q(x) \mid a_r(xq^{-r})a_r(xq^{-r-1}) \dots a_r(xq^{-r-K})$. On a donc :

$$Q(x) \mid \text{pgcd}(a_0(x)a_1(xq) \dots a_0(xq^K), a_r(xq^{-r})a_r(xq^{-r-1}) \dots a_r(xq^{-r-K})).$$

EXEMPLE. $L' = -(qx-1) + (q^5x-1)H_{q,x}^2$, $K = 2$. $Q(x) \mid \text{pgcd}((qx-1)(q^2x-1)(q^3x-1), (q^3x-1)(q^2x-1)(qx-1)) = (qx-1)(q^2x-1)(q^3x-1)$. Donc $x^2(qx-1)(q^2x-1)(q^3x-1)$ est un multiple du dénominateur de toute solution rationnelle de $L = -(qx-1) + q^4(q^5x-1)H_{q,x}^2$.

2.1.2 Borne sur le degré des solutions polynomiales

Soit $L = \sum_{i=0}^n p_i(x)\Delta_{q,x}^i$, avec $p_i \in \mathbb{F}[x]$ pour $i \in \llbracket 0, n \rrbracket$, et soit $d = \max_i(\deg p_i)$. On cherche une borne sur le degré de $y \in \mathbb{F}[x]$ vérifiant $Ly = 0$.

Soit $N \in \mathbb{N}$. On a $\Delta_{q,x}^k x^N = (q^N - 1)^k x^N$, donc si y (que l'on peut supposer unitaire) est de degré N , Ly de degré inférieur à $N + d$, le coefficient de degré $N + d$ étant $\sum_i p_{i,d}(q^N - 1)^i$ (où $p_{i,d}$ est le coefficient de degré d de p_i).

Soit $P(X) = \sum_i p_{i,d}(X-1)^i$. Une condition nécessaire sur N pour que $Ly = 0$ est donc $P(q^N) = 0$. Une borne sur N est donc donnée par $\max(n \geq 0 : P(q^n) = 0)$.

EXEMPLE. $L = ((-q + q^{10})x + 1 - q^{10})\Delta^2 + (-q^{12} + q^{21} - q^{20} - q + q^2 + q^{10})x + ((q^2 - q^{20} - 2q + 2q^{10})x + 1 + q^{20} - 2q^{10})\Delta$ alors :

$$P(X) = q(q-1)(q^2+q+1)(q^6+q^3+1)(X-q)(X-q^{10})$$

et $N \leq 10$.

2.1.3 Calcul de la solution polynomiale

Par coefficients indéterminés On résoud $LP = b$, pour $b \in \mathbb{F}[x]$. Connaissant une borne N sur le degré des solutions polynomiales de l'équation homogène calculée par la méthode de la partie 2.1.2, on considère une solution *a priori* $P = \sum_{i=0}^{\max(N, \deg(b))} a_i X^i$. On calcule LP ; c'est une polynôme dont les coefficients sont des combinaisons linéaires en les a_i . En identifiant coefficient par coefficient dans $LP = b$, on obtient un système linéaire d'inconnues les a_i dont la résolution donne l'ensemble des solutions de l'équation.

Méthode rapide Un inconvénient de la méthode par coefficients indéterminés est qu'elle nécessite la résolution d'un "grand" système linéaire, dont la taille peut être exponentielle en la taille de l'entrée.

Une méthode plus efficace, qui s'appuie sur la récurrence vérifiée par les coefficients de la solution polynomiale, a été introduite dans [1], puis améliorée dans [5, 7]. Pour simplifier, on la présente sur un exemple simple dans le cas différentiel.

Considérons l'équation $(1 + x + x^2)y'(x) - (2xN + N)y(x)$ avec $N \in \mathbb{N}^*$. Par la méthode de la partie 2.1.2, on sait que les solutions polynomiales de cette équation sont de degré au plus $2N$. On vérifie aisément que les coefficients a_n d'une solution série de l'équation vérifient la récurrence $(-2N + n)a_n + (-N + n + 1)a_{n+1} + (n + 2)a_{n+2}$.

On pose $A_n = \begin{bmatrix} a_{n+1} \\ a_n \end{bmatrix}$. On a donc $A_{n+1} = M_n A_n$ avec $M_n = \begin{bmatrix} \frac{N-n-1}{n+2} & \frac{2N-n}{n+2} \\ 1 & 0 \end{bmatrix}$. Une solution série de l'équation différentielle est polynomiale si $A_{2N+1} = 0$. Or $A_{2N+1} = M_0 M_1 \dots M_{2N} A_0$. On calcule donc la matrice $R_{2N} = M_0 M_1 \dots M_{2N}$ (ce calcul peut être effectué efficacement par la méthode dite du scindage binaire), et on résoud le système d'inconnues $a_0, a_1 : R_{2N} \begin{bmatrix} a_1 \\ a_0 \end{bmatrix} = 0$.

Je n'ai pas intégré la méthode rapide à mon code. Celle-ci est plus rapide que la méthode naïve seulement lorsque la solution polynomiale est de grand degré, ce qui n'était pas le cas pour tous les exemples considérés. D'autre part, le temps de résolution de l'équation scalaire étant négligeable devant le temps de découplage, il ne m'a pas semblé nécessaire de chercher à optimiser cette opération.

2.2 Découplage

Le découplage consiste à ramener la résolution d'un système $\partial Y = AY + B$, où $A \in M_n(k(x))$, $B, Y \in k(x)^n$, à celle d'une équation scalaire d'ordre $n : \partial^n y = \sum_{i=0}^{n-1} a_i(x) \partial^i y + b(x)$, avec $a_i, b, y \in k(x)$.

Dans le cas différentiel par exemple, on a le résultat classique suivant, dit du vecteur cyclique (voir par exemple [12]) :

Théorème 1 (vecteur cyclique). *Soit $A \in M_n(\mathbb{Q}(x))$. Il existe $P \in GL_n(\mathbb{Q}(x))$, $c_0, \dots, c_{n-1} \in \mathbb{Q}(x)$ tels que si $Z = (y, y^{(1)}, \dots, y^{(n-1)})$, $Y = (y_1, \dots, y_n)$, $PY = Z$, alors $Y' = AY$ si et seulement si $y^{(n)} = c_{n-1}y^{(n-1)} + \dots + c_0y$.*

2.2.1 Triangularisation du système

La première méthode que j'ai implantée est la triangularisation du système par une base de Gröbner pour l'ordre lexicographique.

La théorie des bases de Gröbner fournit de nombreux algorithmes sur les idéaux d'anneaux de polynômes multivariés (cf. [6] chapitre 15, [13]). Cette théorie se généralise aux algèbres de Ore [11].

On fixe pour toute cette partie $A = k(\mathbf{x})\langle \partial \rangle$ une algèbre de Ore rationnelle, et I un idéal à gauche de A . Les *monômes* de A sont les $\partial^\alpha = \partial_1^{\alpha_1} \dots \partial_n^{\alpha_n}$, pour $\alpha \in \mathbb{N}^n$. On appelle *termes* de A le produit d'une fraction rationnelle non nulle de $k(\mathbf{x})$ par un monôme.

Définition 3 (ordre monomial). Un ordre monomial sur A est un ordre $>$ sur \mathbb{N}^n (et par extension sur l'ensemble des monômes de A par $\alpha > \beta \Leftrightarrow \partial^\alpha > \partial^\beta$) tel que :

- $>$ est un ordre total
- si $\alpha > \beta$, alors $\alpha + \gamma > \beta + \gamma$ pour tous $\alpha, \beta, \gamma \in \mathbb{N}^n$.
- $>$ est un bon ordre sur \mathbb{N}^n (il n'existe pas de suite strictement décroissante infinie).

EXEMPLE. Soient $\alpha, \beta \in \mathbb{N}^n$.

- Ordre lexicographique : $\alpha >_{lex} \beta$ si dans le vecteur $\alpha - \beta$ l'élément non nul le plus à gauche est positif.
- Ordre gradué lexicographique inverse : $\alpha >_{grevlex} \beta$ si $|\alpha| > |\beta|$ ou $|\alpha| = |\beta|$ et dans le vecteur $\alpha - \beta$ l'élément non nul le plus à droite est négatif.

Définition 4 (monôme de tête). Soit $f \in A$ non nul, $f = \sum_{\alpha} c_{\alpha} \partial^{\alpha}$, les c_{α} étant presque tous nuls. Étant donné un ordre monomial $>$, le monôme de tête de f est le monôme ∂^{α} tel que α soit maximal pour $>$ parmi les β tels que $c_{\beta} \neq 0$.

Définition 5 (base de Gröbner). Un ensemble fini $G \subseteq A$ est une base de Gröbner de I si G est un système générateur de I , et si la partie stable du monoïde des monômes de A engendrée par les monômes de tête des éléments de G est égale à celle engendrée par ceux de I .

EXEMPLE. On reprend l'exemple de la partie 1.2. $A = \mathbb{Q}(n, x)\langle \Delta_n, D_x \rangle$, I l'idéal engendré par $\Delta_n^2 + 2(1-x)\Delta_n + 2(1-x)$ et $D_x - G(n, x) + \frac{n}{1-x^2}\Delta_n$, avec $G(n, x) = \frac{-x^3 + nx^2 - (n-1)x + 1}{x(1-x^2)}$. Alors $\{x^4 - x^3 - x^2 - 2x - 2 + (2x^2 + 2x - 2x^4 + x^3)D_x + (x^4 - x^2)D_x^2 - n^2x^2, -x + x^3 - 1 + (x - x^3)D_x + (-x^2 + x)n + nx\Delta_n\}$ est une base de Gröbner de I pour l'ordre lexicographique avec $\Delta_n > D_x$.

Le théorème d'élimination est l'ingrédient essentiel du découplage par bases de Gröbner :

Théorème 2 (élimination). Soient A un anneau, x_1, \dots, x_n des indéterminées, I un idéal à gauche de $A[\mathbf{x}]$. Soit G une base de Gröbner de I pour l'ordre lexicographique $x_1 > x_2 > \dots > x_n$. Alors pour tout $l \in \llbracket 1, n \rrbracket$, $G \cap A[x_l, x_{l+1}, \dots, x_n]$ est base de Gröbner de $I \cap A[x_l, x_{l+1}, \dots, x_n]$.

On introduit n variables formelles g_1, \dots, g_n représentant les variables y_1, \dots, y_n , et d variables h_1, \dots, h_d représentant η_1, \dots, η_d . On considère l'idéal à gauche de l'anneau de polynômes $\mathbb{Q}[\partial, g_1, \dots, g_n, h_1, \dots, h_d]$ engendré par les $\partial y_i - \sum_{j=1}^n A_{i,j} y_j - \sum_{j=1}^d B_{i,j} h_j$ pour $i \in \llbracket 1, n \rrbracket$.

La calcul d'une base de Gröbner pour l'ordre lexicographique $g_1 > \dots > g_n > h_1 > \dots > h_d > \partial$ nous fournit, par le théorème d'élimination, une équation en g_n et en les h_i . On est donc ramené à une équation scalaire, que l'on peut résoudre par l'algorithme de la partie 2.1.

Il suffit alors de substituer g_n par la valeur calculée, et de réitérer le procédé pour calculer les valeurs de y_{n-1}, \dots, y_1 .

Cette méthode présentait l'avantage d'être la plus simple à implanter si l'on disposait, comme c'est le cas en Maple, d'une implantation de calcul de bases de Gröbner. Cependant, pour des systèmes non triviaux d'ordre supérieur à 6 environ, le calcul de la base de Gröbner pour l'ordre lexicographique devenait trop coûteux en temps et en mémoire.

2.2.2 Méthode d'Abramov-Zima

Le seconde méthode de découplage que j'ai implantée est celle d'Abramov-Zima, décrite de manière incomplète dans [2]. Cette méthode explicite et généralise aux algèbres de Ore la construction d'un vecteur cyclique.

Pour être succinct, nous ignorerons comme dans [2] les cas singuliers, qui ne changent pas fondamentalement l'algorithme, mais obligent à distinguer plus de cas. On se place dans une algèbre de Ore rationnelle $k(x)\langle\partial;\sigma,\delta\rangle$, on veut résoudre le système $\partial Y = AY + B$, $A \in M_n(k(x))$, $B \in k(x)^n$, soit :

$$\begin{cases} \partial Y_1 &= A_{1,1}Y_1 + \cdots + A_{1,n}Y_n + B_1 \\ &\vdots \\ \partial Y_n &= A_{n,1}Y_1 + \cdots + A_{n,n}Y_n + B_n \end{cases} \quad (2)$$

L'algorithme fournit en sortie :

- une équation $\partial^n Y_1 = \sum_{k=0}^{n-1} a_k \partial^k Y_1 + r$, $a_{n-1}, \dots, a_0, r \in k(x)$,
- n équations vérifiées par de nouvelles inconnues Z_1, \dots, Z_n : $Z_1 = Y_1$, $Z_i = \partial Z_{i-1} + \sum_{k=1}^{i-1} \beta_{i,k} Z_k + \tau_i$ pour $i \in \llbracket 2, n \rrbracket$,
- une matrice $T \in GL_n(k(x))$ triangulaire supérieure telle que $TY = Z$.

On commence par éliminer Y_2 du membre de droite de (2). On suppose $A_{1,2} \neq 0$ (sinon il s'agit d'un cas que nous avons précédemment annoncé comme singulier). On introduit les nouvelles variables $Z_1 = Y_1$ et :

$$Z_2 = \sum_{k=2}^n A_{1,k} Y_k. \quad (3)$$

Avec la relation $Y_2 = \frac{1}{A_{1,2}}(Z_2 - \sum_{k=3}^n A_{1,k} Y_k)$, on obtient un nouveau système :

$$\begin{cases} \partial Y_1 &= A_{1,1}Y_1 + Z_2 + B_1 \\ \partial Y_2 &= C_{2,1}Y_1 + C_{2,2}Z_2 + C_{2,3}Y_3 + \cdots + C_{2,n}Y_n + B_2 \\ &\vdots \\ \partial Y_n &= C_{n,1}Y_1 + C_{n,2}Z_2 + C_{n,3}Y_3 + \cdots + C_{n,n}Y_n + B_n \end{cases} \quad (4)$$

Pour éliminer ∂Y_2 de (4), on applique ∂ à (3) :

$$\partial Z_2 = \sum_{k=2}^n \sigma(A_{1,k}) \partial Y_k + \delta(A_{1,k}) Y_k \quad (5)$$

et on remplace $Y_2, \partial Y_2, \dots, \partial Y_n$ par leur valeur donnée par (4). On obtient ainsi un système de la forme :

$$\begin{cases} \partial Y_1 &= A_{1,1}Y_1 + Z_2 + B_1 \\ \partial Z_2 &= D_{2,1}Y_1 + D_{2,2}Z_2 + D_{2,3}Y_3 + \cdots + D_{2,n}Y_n + S_2 \\ \partial Y_3 &= C_{3,1}Y_1 + C_{3,2}Z_2 + C_{3,3}Y_3 + \cdots + C_{3,n}Y_n + B_3 \\ &\vdots \\ \partial Y_n &= C_{n,1}Y_1 + C_{n,2}Z_2 + C_{n,3}Y_3 + \cdots + C_{n,n}Y_n + B_n \end{cases} \quad (6)$$

On remplace ensuite Y_3 par une nouvelle inconnue $Z_3 = \sum_{k=3}^n D_{2,k}Y_k$ en supposant $D_{2,3} \neq 0$, et on itère le procédé jusqu'à l'obtention d'un système de la forme :

$$\begin{cases} \partial Y_1 &= -\beta_{1,1}Y_1 + Z_2 - \tau_1 \\ \partial Z_2 &= -\beta_{2,1}Y_1 - \beta_{2,2}Z_2 + Z_3 - \tau_2 \\ &\vdots \\ \partial Z_n &= -\beta_{n,1}Y_1 - \beta_{n,2}Z_2 \cdots - \beta_{n,n}Z_n - \tau_n \end{cases} \quad (7)$$

Pour obtenir l'équation scalaire en Y_1 à partir de (7), on applique d'abord ∂ à la première équation, ce qui permet d'exprimer ∂Z_2 en fonction de $Y_1, \partial Y_1, \partial^2 Y_1$. En appliquant ∂ à la seconde équation et en remplaçant ∂Z_2 par son expression en $Y_1, \partial Y_1, \partial^2 Y_1$, on obtient l'expression de Z_3 en fonction de $Y_1, \partial Y_1, \partial^2 Y_1, \partial^3 Y_1$. En itérant, on obtient par l'avant-dernière équation l'expression de ∂Z_n en fonction de $Y_1, \partial Y_1, \dots, \partial^n Y_1$; on substitue cette dernière expression au membre de gauche de la dernière équation, et on remplace Z_2, \dots, Z_n dans le membre de droite par leur expression en les $\partial^i Y_1$, ce qui donne l'équation voulue.

Pour des systèmes de petite taille, la méthode d'Abramov-Zima s'avère plus rapide que celle par bases de Gröbner pour l'ordre lexicographique. Cependant, la taille des fractions rationnelles manipulées augmente très rapidement avec la taille des systèmes considérés, rendant impossible la résolution de systèmes de dimension supérieure à 5 ou 6 par cette méthode.

2.2.3 “FGLM à la main”

Comme dans le cas commutatif, le calcul d'une base de Gröbner par l'algorithme de Buchberger est plus ou moins difficile suivant l'ordre monomial choisi. Expérimentalement, il est plus facile de calculer une base de Gröbner pour l'ordre gradué lexicographique inverse que pour l'ordre lexicographique.

Faugère, Gianni, Lazard et Mora présentent dans [14] l'algorithme FGLM qui permet à partir d'une base de Gröbner pour un certain ordre monomial de calculer une base de Gröbner pour un autre ordre. En particulier, cela permet de calculer une base de Gröbner pour l'ordre lexicographique à partir d'une base de Gröbner pour l'ordre gradué lexicographique inverse.

D'autre part, un seul élément de la base de Gröbner nous intéresse : l'équation scalaire. La méthode FGLM commence par calculer cette équation, ce qui nous permet de ne pas calculer toute la base de Gröbner.

On commence par calculer une base de Gröbner de l'idéal donné en partie 2.2.1 pour l'ordre gradué lexicographique inverse. On calcule ensuite la forme normale de $g_n, \partial g_n, \dots, \partial^n g_n$ modulo cet idéal. Par un argument de dimension, il existe $\lambda_0, \dots, \lambda_n \in \mathbb{Q}(x)$ tels que les composantes en g_1, \dots, g_n de $\sum_{k=0}^n \lambda_k \partial^k g_n$ modulo l'idéal soient nulles. Autrement dit, on a obtenu une combinaison linéaire des $\partial^i g_n$ qui ne dépend pas des g_i : il s'agit de l'équation scalaire.

Cette méthode est la seule qui nous a permis de résoudre des systèmes de taille supérieure à 10. L'algorithme de création télescopique sur l'exemple de la partie 1.4.1 génère une système différentiel d'ordre 16, que l'implantation de cette dernière méthode résout en environ 40s.

2.3 Approche directe

La méthode de résolution directe par Barkatou [4] semblait trop longue à implanter durant le stage, et l'implantation de l'auteur n'était pas utilisable, car elle ne supportait pas de corps

de fractions rationnelles comme corps de base, et n'était pas facilement modifiable. On donne une description de cette méthode dans le cas différentiel.

Pour $a \in k(x)$, $a \neq 0$, on notera $\text{val}_\infty(a) \in \mathbb{Z}$ la valuation du développement à l'infini de a en série formelle de Laurent en $1/x$. On note $\text{lc}_\infty(a)$ le coefficient en $x^{-\text{val}_\infty(a)}$ de ce développement. On fera parfois l'abus d'appeler $-\text{val}_\infty(a)$ le degré de a .

On veut résoudre le système :

$$\mathcal{M}(Y) = x \frac{dY}{dx} - M(x)Y = N(x) \quad (8)$$

pour $Y \in k(x)^n$ ($A \in M_n(k(x))$, $B \in k(x)^n$).

Par une méthode analogue à celle du cas scalaire on calcule $R \in k[x]$ tel que pour tout Y solution de (8), $RY \in k[x]^n$. On pose alors $Z = RY$; on a $x \frac{dZ}{dx} = x \frac{dR}{dx} Y + xR \frac{dY}{dx} = x \frac{dR}{dx} \frac{Z}{R} + R(MY + N) = (x \frac{dR}{dx} \frac{1}{R} + M)Z + RN$, on est donc ramené à résoudre une équation du même type que (8) en $Z \in k[x]^n$.

On cherche une solution polynomiale de (8). On commence par calculer le coefficient dominant et le degré de Y : $c \in k^n$, $\mu \in \mathbb{N}$ tels que $Y = cx^\mu + Z$, avec $\mu = \deg(Y) > \deg(Z)$. Pour se faire, la méthode de Barkatou s'appuie sur le développement à l'infini des solutions.

On commence par éliminer les éventuelles singularités de M en l'infini. Pour cela, soient $\alpha_i = \min(\max_j(\deg(M_{i,j})), 0)$, et $D = \text{Diag}(x^{-\alpha_1}, \dots, x^{-\alpha_n})$. Soient $A = DM$ et $B = DN$. Le système (8) est donc équivalent au système :

$$\mathcal{L}(Y) = xD \frac{dY}{dx} - A(x)Y = B(x). \quad (9)$$

EXEMPLE. On prend $M = \begin{bmatrix} 5 & \frac{1}{x} \\ \frac{1}{1+x} & -\frac{1}{3}(1+x^2) \end{bmatrix}$ et $N = \begin{bmatrix} \frac{1}{x} + 5 + 7x + 6x^2 + 4x^3 + x^4 \\ -\frac{2}{3} + \frac{16}{3}x^2 \end{bmatrix}$. On a $D = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{x^2} \end{bmatrix}$, $A = \begin{bmatrix} 5 & \frac{1}{x} \\ \frac{1}{x^2(x+1)} & -\frac{1}{3}(\frac{1}{x^2} + 1) \end{bmatrix}$, $B = \begin{bmatrix} \frac{1}{x} + 5 + 7x + 6x^2 + 4x^3 + x^4 \\ -\frac{2}{3}\frac{1}{x^2} + \frac{16}{3} \end{bmatrix}$.

A admet un développement en série en x^{-1} en l'infini, et D est un polynôme en x^{-1} , à coefficients dans $M_n(k)$; on note $A_0, A_1 \dots$ (respectivement D_0, D_1, \dots) les coefficients de A dans ce développement. On étudie l'action de \mathcal{L} sur le monôme cx^μ . On a :

$$\mathcal{L}(cx^\mu) = x^\mu(D\mu - A) = x^\mu((D_0\mu - A_0)c + \dots) \quad (10)$$

où les \dots désignent des termes en x^{-1} de degré supérieur à 1. Donc $\deg(\mathcal{L}(cx^\mu)) \leq \mu$, et si $\deg(B) < \mu$, alors nécessairement $(D_0\mu - A_0)c = 0$. Les valeurs de μ pour lesquelles $(D_0\mu - A_0)$ n'est pas inversible ont un rôle essentiel. On introduit donc les définitions suivantes :

Définition 6 (polynôme indiciel à l'infini ; simplicité à l'infini). Soit $\mathcal{M}(Y) = x \frac{dY}{dx} - M(x)Y$; on définit D, A comme précédemment. On appelle *polynôme indiciel* de \mathcal{M} à l'infini, noté E_∞ , le polynôme de $k[\lambda]$: $\det(D_0\lambda - A_0)$. L'opérateur \mathcal{M} est dit *simple à l'infini* si $E_\infty \neq 0$.

EXEMPLE. On prend M de l'exemple précédent. $D = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} x^{-2}$, $A = \begin{bmatrix} 5 & 0 \\ 0 & -\frac{1}{3} \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x^{-1} + \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{3} \end{bmatrix} x^{-2} + \dots$. Donc $E_\infty = \det \left(\begin{bmatrix} \lambda - 5 & 0 \\ 0 & -\frac{1}{3} \end{bmatrix} \right) = \frac{1}{3}(5 - \lambda)$.

D'après un résultat de [16], il est toujours possible de se ramener à un opérateur simple :

Proposition 3. *Étant donné un système différentiel $\mathcal{M}(Y) = N$, il existe $T \in GL_n(k(x))$ telle que si $Y = TZ$, alors Z vérifie un système différentiel équivalent $\tilde{\mathcal{M}}(Z) = \tilde{N}$ simple à l'infini. De plus la matrice T^{-1} est polynomiale en x .*

Soient $\delta = \text{val}_\infty(B)$, $b_\delta = \text{lc}_\infty(B)$. On a :

$$\mathcal{L}(Z) = b_\delta x^\delta + x^\mu(-\mu D_0 + A_0)c + \{\text{termes de degré} < \max(\delta, \mu)\}. \quad (11)$$

Soient $\mathcal{R} = \{\lambda \in \mathbb{N} : E_\infty(\lambda) = 0\}$, $r = \max(\mathcal{R})$. Il faut que $\deg(\mathcal{L}(Z)) \leq \deg(Z) < \mu$, on a donc trois cas possibles :

1. Si $\mathcal{R} = \emptyset$ et $\delta < 0$, pas de solutions.
2. Si $r > \delta$, on prend $\mu = r$ et $c \in \ker(-rD_0 + A_0)$.
3. Si $r \leq \delta$ ou $\mathcal{R} = \emptyset$ et $\delta \geq 0$. Alors il faut prendre $\mu = \delta$. L'équation (11) devient donc $\mathcal{L}(Z) = x^\delta(b_\delta - (\delta D_0 - A_0)c) + \{\text{termes de degré} < \delta\}$. Donc si $b_\delta \notin \text{Im}(\delta D_0 - A_0)$, il n'y a pas de solutions. Sinon, on prend c tel que $(\delta D_0 - A_0)c = b_\delta$.

EXEMPLE. M, N des exemples précédents. $\delta = 4$, $b_\delta = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\mathcal{R} = \{5\}$, $r = 5$. On est dans le cas 2, donc $\mu = 5$, et $c \in \ker\left(\begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{3} \end{bmatrix}\right) = \text{Vect}\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right)$, donc $c = \begin{bmatrix} a_1 \\ 0 \end{bmatrix}$ avec a_1 constante.

Connaissant c et μ , on revient à l'équation (9), qui devient $\mathcal{L}(Z) = B - \mathcal{L}(cx^\mu)$. On itère le procédé jusqu'à obtenir un système de la forme $\mathcal{L}(W) = F = B - \mathcal{L}(c_1x^{\mu_1} + \dots + c_lx^{\mu_l})$ qui n'a pas de solution polynomiale non nulle de degré $< \mu_l$. L'équation (9) a des solutions polynomiales si et seulement si $F = 0$. F est un vecteur de fractions rationnelles qui dépendent linéairement des c_i . On pose $F = 0$, ce qui donne un système linéaire en les c_i . Les solutions de ce système donnent donc les solutions polynomiales de (9) (et donc de (8)).

EXEMPLE. M, N des exemples précédents. Après itération de l'algorithme, on obtient :

$$\begin{aligned} F &= B - \mathcal{L}\left(\begin{bmatrix} a_1 \\ 0 \end{bmatrix} x^5 + \begin{bmatrix} 1 \\ 0 \end{bmatrix} x^4 + \begin{bmatrix} 2 \\ 0 \end{bmatrix} x^3 + \begin{bmatrix} 2 \\ 3a_1 \end{bmatrix} x^2 + \begin{bmatrix} \frac{1}{4}(7 - 3a_1) \\ 3 - 3a_1 \end{bmatrix} x + \begin{bmatrix} \frac{1}{5}(2 + 3a_1) \\ 19 - 18a_1 \end{bmatrix} x^0\right) \\ &= \begin{bmatrix} \frac{18-18a_1}{x} \\ 1/20 \frac{(60a_1-60)x^2+(185a_1-185)x+132a_1-132}{x^2(x+1)} \end{bmatrix} \end{aligned}$$

On a donc $F = 0$ si et seulement si $a_1 = 1$, et l'unique solution de l'équation (8) est donc :

$$Y = \begin{bmatrix} x^5 + x^4 + 2x^3 + 2x^2 + x + 1 \\ 3x^2 + 1 \end{bmatrix}.$$

3 Exploitations

3.1 Transformée de Laplace

Définition 7 (transformée de Laplace). Soit $f : \mathbb{R}^+ \rightarrow \mathbb{C}$. Sa transformée de Laplace est définie par :

$$F(s) = \mathcal{L}\{f(t)\} = \int_0^{\infty} e^{-st} f(t) dt.$$

Si f est solution d'une équation différentielle linéaire à coefficients polynomiaux, alors $(s, t) \mapsto e^{-ts} f(t)$ est ∂ -finie. On peut donc utiliser l'algorithme de création télescopique pour calculer une équation différentielle vérifiée par F .

3.2 Coefficients de développements en série de Tchebycheff

Pour cette partie, on fixe l'algèbre de Ore rationnelle $A = \mathbb{Q}(n, x) \langle \Delta_n, D_x \rangle$.

Définition 8 (polynôme de Tchebycheff de première espèce). Pour tout $n \in \mathbb{N}$, il existe un unique polynôme $T_n \in \mathbb{Z}[X]$ tel que $\cos(nx) = T_n(\cos(x))$ pour tout $x \in \mathbb{C}$. T_n est le n -ème polynôme de Tchebycheff de première espèce.

Proposition 4. Pour tous $n \in \mathbb{N}^*$, $x \in \mathbb{C}$ on a $T_{n+2}(x) = 2xT_{n+1}(x) - T_n(x)$ et $T'_n(x) = \frac{n}{1-x^2}(xT_n(x) - T_{n+1}(x))$. La famille $(T_n(x))_n$ est donc ∂ -finie sous l'action de A .

Proposition 5. $(T_n(x))_n$ est une famille orthogonale de $L^2\left([-1, 1], \frac{dx}{\sqrt{1-x^2}}\right)$.

Définition 9 (développement en série de Tchebycheff). On appelle développement en série de Tchebycheff à l'ordre n d'une fonction f de $L^2([-1, 1])$ la projection orthogonale de f sur T_0, \dots, T_{n-1} .

Le n -ème terme du développement est donc égal à $\frac{2}{\pi} \int_{-1}^1 \frac{f(t)T_n(t)}{\sqrt{1-t^2}} dt$. Si f est solution d'une équation différentielle linéaire à coefficient polynomiaux, $(n, x) \mapsto \frac{f(x)T_n(x)}{\sqrt{1-x^2}}$ est ∂ -finie sous l'action de A , et notre algorithme de création télescopique permet donc de calculer une récurrence sur les coefficients de la série de Tchebycheff.

Contrairement à la plupart des algorithmes dédiés actuels, notre algorithme permet par exemple de retrouver la récurrence d'ordre minimale sur les coefficients du développement de $(1+x)J_0(2\sqrt{1-x^2})$ en série de Tchebycheff (cf. [18]) :

```
> f := (1+x)*BesselJ(0, 2*sqrt(1+x)):
> Mgfun:-creative_telescoping(2/Pi*f*ChebyshevT(n,x)/sqrt(1-x^2),
n::shift, x::diff)[1][1];
```

$$(-1/2n - 1) _F(n) - 1/2 \frac{(2n^5 + 12n^4 + 23n^3 + 9n^2 - 17n - 13) _F(n+1)}{n^2 + 4n + 4}$$

$$- 1/2 \frac{(2n^4 + 14n^3 + 31n^2 + 28n + 8) _F(n+2)}{n+2} - 1/2 \frac{(3n^2 + n^3 + 3n + 1) _F(n+3)}{n^2 + 4n + 4}$$

Conclusion. Ce qu'il reste à faire.

Implantation

Mon implantation a permis de prouver automatiquement toutes les identités qui l'avaient été par les précédentes implantations, ainsi que nouvelles identités qui n'avaient jusqu'alors jamais été prouvées autrement que par une utilisation interactive ou experte de ces algorithmes.

Ma fonction `iterated_creative_telescoping` ne gère actuellement que le cas où l'opérateur Q s'annule aux limites de l'intervalle d'intégration. Pour automatiser entièrement le calcul de certaines sommes et intégrales multiples, il serait nécessaire de garder traces de certaines étapes du calcul.

Il reste à ajouter l'implantation de l'algorithme de Chyzak, Kauers et Salvy de 2009. Cette implantation ne faisait pas partie de l'objectif minimal du stage, et une importante réorganisation de mon code aurait été nécessaire pour l'ajouter *a posteriori*, ce qui n'aurait pas été possible dans les dernières semaines du stage. De plus mon implantation exploite la fonctionnalité de Mgfund qui fournit le système d'équations vérifié par une fonction ∂ -finie, qu'il sera nécessaire d'étendre au cas non ∂ -fini.

Théorie

Solutions rationnelles de systèmes différentiels

Il m'est apparu au cours du stage que le problème du calcul des solutions polynomiales et rationnelles de systèmes différentiels linéaires $Y'(x) = A(x)Y(x) + B(x)$ ($A \in M_n(k(x))$, $B \in k(x)^n$) méritait d'être approfondi.

Un algorithme de complexité quasi-linéaire en la taille de la sortie est connu pour la résolution polynomiale pour le cas scalaire $y' = ay + b$ ($a, b \in k(x)$) depuis Brent et Kung [8]; cet algorithme est très simple, et s'appuie sur les techniques de calculs rapides sur des séries par la méthode d'accélération de convergence de Newton. Dans le cas des équations scalaires $y^{(n)} = a_{n-1}(x)y^{(n-1)} + \dots + a_0(x)y(x) + b(x)$ on dispose d'un algorithme simple et efficace avec une étude précise de la complexité par Bostan, Cluzeau et Salvy dans [7]; ici encore l'algorithme est de complexité quasi-linéaire en le degré de la solution.

Cependant, dans le cas des systèmes, il n'existe pas à l'heure actuelle d'algorithme satisfaisant. Les techniques de découplage, s'appuyant sur le vecteur cyclique, font "exploser" la taille des fractions rationnelles manipulées. D'autre part, cette technique n'exploite pas que l'on cherche des solutions polynomiales ou rationnelles. De plus on ne dispose pas de résultats théoriques permettant de quantifier la taille des données manipulées. L'algorithme de Barkatou [4], qui m'a semblé généraliser les idées de la méthode dans le cas scalaire, m'a paru être une meilleure approche, mais moins facile à mettre en œuvre car significativement plus longue à implanter.

Concrètement, la résolution polynomiale d'un système différentiel non trivial d'ordre supérieur à 20 est actuellement quasi inextricable pour toutes les implantations dont nous disposons.

Création télescopique

La création télescopique reposant sur la résolution rationnelle, il n'y a *a fortiori* pas de résultat de complexité sur cette opération.

Les preuves actuelles de terminaison de l'algorithme et de majoration de l'ordre de l'opérateur retourné par l'algorithme s'appliquent aux fonctions holonomes, qui est une sous-classe des fonctions ∂ -finies. Il serait donc intéressant d'étendre ces résultats aux fonctions ∂ -finies.

Remerciements

Je tiens tout d'abord à remercier chaleureusement les équipes du Laboratoire joint INRIA – Microsoft Research d'Orsay et de l'équipe projet ALGORITHMS de l'INRIA Rocquencourt pour leur accueil, pour l'ambiance amicale qui règne dans ces deux laboratoires, et pour les conditions de travail idéales qui m'ont ainsi été fournies. Je remercie notamment ceux avec qui j'ai eu l'occasion d'avoir des conversations et des réflexions enrichissantes, Alexandre, Alin, Bruno, Christoph, Marc, et Stephan. Je remercie enfin tout particulièrement Frédéric Chyzak pour avoir accepté de m'encadrer durant ce stage, pour sa disponibilité et son écoute, et pour avoir eu la patience de répondre à toutes mes questions pas toujours profondes ni pertinentes...

Références

- [1] Sergei A. Abramov, Manuel Bronstein, and Marko Petkovšek. On polynomial solutions of linear operator equations. In A. Levelt, editor, *Symbolic and algebraic computation*, pages 290–296, New York, 1995. ACM Press.
- [2] Sergei A. Abramov and Eugene V. Zima. A universal program to uncouple linear systems. In *Proceedings of CMCP'96 (International Conference on Computational Modeling and Computing in Physics, Dubna, Russia)*, volume 7, pages 16–26, 1996.
- [3] Milton Abramowitz and Irene A. Stegun, editors. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. Dover Publications Inc., New York, 1973. Reprint of the 1972 edition.
- [4] Moulay A. Barkatou. On rational solutions of systems of linear differential equations. *Journal of symbolic computation*, 28(4-5) :547–567, 1999.
- [5] Alin Bostan, Frédéric Chyzak, Thomas Cluzeau, and Bruno Salvy. Low complexity algorithms for linear recurrences. In Jean-Guillaume Dumas, editor, *ISSAC'06*, pages 31–38. ACM Press, 2006.
- [6] Alin Bostan, Frédéric Chyzak, Marc Giusti, Romain Lebreton, Bruno Salvy, and Éric Schost. Algorithmes en calcul formel et en automatique - notes du cours 2-22 du MPRI, année 2008-2009, 2009. <http://algo.inria.fr/salvy/mpri/poly.pdf>.
- [7] Alin Bostan, Thomas Cluzeau, and Bruno Salvy. Fast algorithms for polynomial solutions of linear differential equations. In Manuel Kauers, editor, *ISSAC'05*, pages 45–52, New York, 2005. ACM Press. Proceedings of the 2005 International Symposium on Symbolic and Algebraic Computation, July 2005, Beijing, China.
- [8] R. P. Brent and H. T. Kung. Fast algorithms for manipulating formal power series. *Journal of the ACM*, 25(4) :581–595, 1978.
- [9] Frédéric Chyzak. An extension of Zeilberger's fast algorithm to general holonomic functions. *Discrete Mathematics*, 217(1-3) :115–134, 2000.
- [10] Frédéric Chyzak, Manuel Kauers, and Bruno Salvy. A non-holonomic systems approach to special functions identities. 2009.
- [11] Frédéric Chyzak and Bruno Salvy. Non-commutative elimination in Ore algebras proves multivariate identities. *Journal of symbolic computation*, 26(2) :187–227, 1998.
- [12] Frances Thorndike Cope. Solutions of irregular linear differential equations. part ii. *American Journal of Mathematics*, 58(1) :130–140, January 1936.
- [13] David Cox, John Little, and Donal O'Shea. *Ideals, varieties, and algorithms*. Springer-Verlag, New York, second edition, 1996.
- [14] J.-C. Faugère, P. Gianni, D. Lazard, and T. Mora. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *Journal of symbolic computation*, 16 :329–344, October 1993.
- [15] M. L. Glasser and E. Montaldi. Some integrals involving Bessel functions. *Journal of mathematical analysis and applications*, 183(3) :577–590, May 1994.
- [16] A. Hilali and A. Wazner. Formes super-irréductibles des systèmes différentiels linéaires. *Numerische Mathematik*, 50(4) :429–449, July 1987.

- [17] Oystein Ore. Theory of non-commutative polynomials. *Annals of Mathematics*, 34 :480–508, 1933.
- [18] Luc Rebillard and H. Zakrajšek. Recurrence relations for the coefficients in hypergeometric series expansions. In Ilias Kotsireas and Eugene Zima, editors, *Computer Algebra 2006*, Latest Advances in Symbolic Algorithms, pages 158–180. World Scientific, 2006.

Annexe 1 : Expérimentation

Voici quelques tests sur lesquels j'ai testé mon implantation `creative_telescoping`. Le code a été exécuté avec Maple 13 sur un MacBook 2.4GHz. La colonne dimension contient la dimension de l'espace vectoriel $V = \text{Vect}(\partial^\alpha f)$, la colonne Ordre P contient l'ordre de l'opérateur P retourné par l'algorithme (la plus grande valeur d testée par l'algorithme), qui est aussi égale au nombre de systèmes linéaires résolus par le programme.

Intégrale, somme	Dimension	Ordre P	Temps / Mémoire
$\sum_m \frac{(a)_{m+n}(b)_{m+n}}{m!n!(c)_m(d)_n} x^m y^n$	1	4	0.264s / 20.84Mo
$\int t K_0(at) J_0(bt) dt$	4	1	0.301s / 20.71Mo
$\int t^{2k+1} I_0(t) K_0(t)^4 dt$	6	3	0.778s / 87.93Mo
$\sum_n J_n(x) y^n$	2	1	0.104s / 9.57Mo
$\sum_n J_{2n+1/2}(x)^2$	2	1	0.127s / 10.09Mo
$\sum_n J_n(z)^2$	3	1	0.195s / 13.63Mo
$\int x e^{-px^2} J_n(bx) I_n(cx) dx$	4	2	0.335s / 27.52Mo
$\int x e^{-px^2} J_\nu(ax) J_\nu(bx) dx$	4	2	0.329s / 27.13Mo
$\int x e^{-\eta x^2} J_b(Kx) Y_b(kx) dx$	4	2	0.334s / 27.52Mo
$\int x J_1(ax) I_1(ax) Y_0(x) K_0(x) dx$	16	1	46.416s / 129.48Mo
$\int e^{-x} e^{-y} J_1(x) J_1(y) J_2(c\sqrt{xy}) dx$	8	2	4.419s / 78.63Mo
$\int x^a e^{-bx} K_m(cx) K_n(dx) dx$	4	6	12.113s / 91.47Mo
$\int x^a e^{-bx} K_m(cx) K_n(dx) dx$	4	6	12.160s / 88.20Mo
$\int x^a e^{-bx} K_m(cx) K_n(dx) dx$	4	4	1.668s / 81.25Mo
$\int x^a e^{-bx} K_m(cx) K_n(dx) dx$	4	4	1.669s / 81.12Mo
$\int x^a e^{-bx} K_m(cx) K_n(dx) dx$	4	4	1.181s / 82.04Mo
$\int x^a e^{-bx} K_m(cx) K_n(dx) dx$	4	4	1.180s / 82.04Mo
$\sum_k \binom{n}{k}$	1	1	0.096s / 9.04Mo
$\sum_k \binom{n}{k}^2$	1	1	0.123s / 13.50Mo
$\sum_k \binom{n}{k}^3$	1	2	0.200s / 23.33Mo
$\sum_k \binom{n}{k}^4$	1	2	0.295s / 39.71Mo
$\sum_k \binom{n}{k}^5$	1	3	1.187s / 80.86Mo
$\sum_k (-1)^k \binom{n}{k} \binom{2k}{n}$	1	1	0.149s / 14.68Mo
$\sum_k (-1)^k \binom{n}{k} \binom{3k}{n}$	1	2	0.204s / 23.46Mo
$\sum_k (-1)^k \binom{n}{k} \binom{4k}{n}$	1	3	0.342s / 34.47Mo
$\sum_k (-1)^k \binom{n}{k} \binom{5k}{n}$	1	4	0.642s / 81.38Mo
$\sum_k (-1)^k \binom{n}{k} \binom{6k}{n}$	1	5	1.577s / 97.50Mo
$\sum_s \frac{2^{(i-r)(1+2(n-j-s))}}{(2i-r)(2(n-j)-s+1)} \binom{2i-1}{r} \binom{2(j-i)-1}{2j-n-r+s-1} \binom{2(n-j)}{s}$	1	1	0.973s / 89.90Mo
$\sum_i \binom{i+j}{j}^2 \binom{4n-2i-2j}{2n-2j}$	1	2	45.842s / 195.52Mo
$\sum_s (-1)^{n+r+s} \binom{n}{r} \binom{n}{s} \binom{n+r}{r} \binom{n+s}{s} \binom{2n-r-s}{n}$	1	2	9.513s / 103.00Mo
$\int \frac{e^{-px} T_n(x)}{\sqrt{1-x^2}} dx$	2	2	0.184s / 15.73Mo
$\int \frac{x e^{px} T_n(x)}{\sqrt{1-x^2}} dx$	2	2	0.205s / 17.17Mo
$\int (1+x) J_0(2\sqrt{x+1}) \frac{T_n(x)}{\sqrt{1-x^2}} dx$	4	3	0.935s / 83.87Mo

Intégrale, somme	Dimension	Ordre P	Temps / Mémoire
$\int \frac{\sqrt{y(y-1)x^4+2yx^2+2}}{(yx^2+1)(x^2+1)} dx$	1	4	1.065s / 78.37Mo
$\int e^{-xt^2} dt$	1	1	0.055s / 5.11Mo
$\int (1-y-x(1-y)y)^{-1} dy$	1	1	0.087s / 7.60Mo
$\int (x(2a-x))^{\nu-1/2} C_n^{(\nu)}\left(\frac{x}{a}-1\right) e^{-bx} dx$	2	2	0.238s / 20.31Mo
$\int \frac{(1+2xy+4y^2) \exp\left(\frac{4x^2y^2}{1+4y^2}\right)}{y^{n+1}(1+4y^2)^{\frac{3}{2}}} dx$	1	2	0.239s / 17.04Mo
$\int x^{r-1}(1-x)^{s-1} {}_2F_1(a, b, c, x) dx$	2	2	0.560s / 40.62Mo
$\int \left(1 - 4 \frac{w^2}{(1-2wt+\sqrt{1-4wt+4w^2t^2-4w^2})^2}\right)^{-1} dt$	2	1	0.293s / 21.89Mo
$\sum_n P_n^{(a,b)}(x)y^n$	2	2	1.528s / 75.88Mo
$\sum_n P_n(x)y^n$	2	1	0.163s / 15.07Mo
$\int \text{RootOf}(t_Z^2x + (t+tx^2-x)_Z + tx) dt$	2	2	0.442s / 32.50Mo
$\int \frac{1}{(x^4+2ax^2+1)^{m+1}} dx$	1	2	0.131s / 16.38Mo
$\sum_k \frac{\binom{m}{k}(n+k)!}{z^n z^k} - \frac{(-1)^{n+m} \binom{n}{k}(m+k)!}{z^m z^k}$	2	1	1.446s / 72.21Mo
$\int \frac{1}{1-x-y} dx$	1	1	0.060s / 5.50Mo
$\int \frac{1}{1-x-y-xy} dx$	1	1	0.072s / 6.03Mo
$\int \frac{1}{1-x-xy(1-x)^2} dx$	1	2	0.129s / 9.96Mo
$\int \frac{1}{1-x-xy(1-x)^3} dx$	1	3	0.255s / 18.48Mo
$\int \frac{1}{1-x-xy(1-x)^4} dx$	1	4	0.700s / 45.87Mo
$\int \frac{1}{1-x-xy(1-x)^5} dx$	1	5	2.056s / 79.55Mo
$\int \frac{1}{1-x-xy(1-x)^6} dx$	1	6	5.596s / 79.42Mo
$\int \frac{1}{1-x-xy(1-x)^7} dx$	1	7	14.420s / 97.63Mo
$\int \frac{xy(1-x)^3}{(1-x)^4-xy(1-x-x^2+x^3+x^2y)} dx$	1	4	1.909s / 78.24Mo
$\int \text{Ei}\left(-\frac{t}{x}\right) Y_1(cx) dx$	4	4	0.510s / 35.78Mo
$\int J_0(b\sqrt{x+a})^2 P_n\left(\frac{x}{a}\right) dx$	6	3	20.812s / 144.81Mo
$\sum_k \frac{1}{[n-k]![n+k]!}$	1	2	0.289s / 88.59Mo
$\int \frac{\cos(zt)}{\sqrt{1-t^2}} dt$	2	2	0.164s / 11.79Mo
$\int \frac{\sin(zt)}{\sqrt{t^2-1}} dt$	2	2	0.163s / 11.66Mo
$\int e^{-tz} \sin\left(\frac{z}{1+z^2}\right) dz$	2	8	0.749s / 46.39Mo
$\int \frac{1}{(x^2+1)^{m+1}} dx$	1	1	0.070s / 8.78Mo