Bruno Gauthier
gauthier@univ-mlv.fr

DEA Informatique Fondamentale et Applications
Université Marne La Vallée
Année universitaire 1995–1996

Rapport de Stage de DEA

Séries hypergéométriques

Démonstration automatique

Directeur de stage : Jacques Désarménien

Laboratoire d'accueil : Institut Gaspard Monge

Table des matières

Introduction

L'étude des séries hypergéométriques basiques (aussi appelées séries q-hypergéométriques ou q-séries) a commencé essentiellement en 1748 quand Euler a considéré le produit infini $(q;q)_\infty^{-1} = \prod_{k=0}^{\infty}(1-q^{k+1})^{-1}$ comme la fonction génératrice de $p(n)$, le nombre de partitions d'un entier positif n.

Dans les dernières années, les séries q-hypergéométriques ont été étudiées en physique et en statistiques, mais aussi dans des domaines plus familiers aux q-séries tels que la combinatoire et la théorie des nombres. Ces études ont notamment été enrichies par l'apport des q-analogues des fonctions spéciales.

Dans les différents chapitres de ce rapport, nous allons nous intéresser à des identités hypergéométriques difficiles à démontrer par les techniques mathématiques traditionnelles. Les ordinateurs permettent de le faire et ceci sans aucune hypothèse et très rapidement pour une certaine classe d'identités. De plus, la vérification de la preuve fournie est une formalité. En effet, les méthodes proposées permettent de vérifier la validité des résultats retournés par la machine.

Nous nous intéressons ici en particulier aux récurrences linéaires à coefficients polynomiaux satisfaites par des séries formelles. Ces récurrences permettent la démonstration d'identités non triviales faisant intervenir par exemple des coefficients binomiaux. La méthode utilisée pour démontrer une égalité du type $\sum f = \sum g$ est la suivante :

- On construit les équations de récurrence satisfaites par les deux membres de l'équation.
- On vérifie que ces deux équations de récurrence se résument à une unique récurrence satisfaite par les deux membres.
- On vérifie également que des conditions initiales en nombre suffisant sont identiques pour les deux membres de l'égalité.

Remarque : la récurrence trouvée donne également des informations sur la série (notamment sur son comportement asymptotique) sans avoir besoin de calculer explicitement cette série.

Doron Zeilberger a étudié ce problème et a mis en place un algorithme qui permet de répondre à certaines interrogations. La première partie de ce rapport est consacrée à la description de cet algorithme, la seconde partie concerne l'extension aux q-séries. Enfin, la troisième partie expose une autre approche faisant appel à une théorie algébrique plus complexe.

Je tiens à remercier Frédéric Chyzak [Projet Algorithmes de l'I.N.R.I.A.] pour ses précieux conseils durant mes recherches ainsi que mon directeur de stage Jacques Désarménien pour son soutien.

Chapitre 1

Notations et Définitions

1.1 Notations usuelles

Introduisons quelques fonctions usuelles :

$\Gamma(z) := \int_0^\infty t^{z-1}\, e^{-t}dt, \qquad \Gamma(n+1) = n!$		fonction Gamma
$(a)_k := a\,(a+1)\ldots(a+k-1), \qquad k = 1, 2, \ldots;$	$(a)_0 := 1;$	factorielle montante
$(a;q)_k := \prod_{j=0}^{k-1}(1 - aq^j), \qquad k = 1, 2, \ldots;$	$(a;q)_0 := 1;$	q-factorielle montante
$(a_1, \ldots, a_r; q)_k := (a_1;q)_k\,(a_2;q)_k \ldots (a_r;q)_k$		

1.2 Définitions élémentaires

1.2.1 Préliminaires

Définition 1 *Une fonction $F(k)$ est* ***rationnelle*** *en k s'il existe deux polynomes s et t premiers entre eux tels que $F(k) = s(k)/t(k)$.*

Définition 2 *Une fonction $F(k)$ est* ***géométrique*** *en k si le rapport $F(k+1)/F(k)$ est constant.*

Définition 3 *Une série $\sum_{k\geq 0} F(k)$ est* ***géométrique*** *si le rapport entre deux termes consécutifs est constant ($F(k+1)/F(k) = \texttt{const}$).*

Définition 4 *Une fonction $G(k)$ est* ***hypergéométrique*** *en k si le rapport $G(k+1)/G(k)$ est une fonction rationnelle de k à coefficients dans $\mathbb{C}$.*

Définition 5 *Une série $\sum_{k\geq 0} G(k)$ est* ***hypergéométrique*** *si le rapport entre deux termes consécutifs est une fonction rationnelle de l'indice de sommation k :*
$G(k+1)/G(k) = P(k)/Q(k)$ où P et Q sont des polynômes en k à coefficients dans $\mathbb{C}$.

Définition 6 *De façon analogue, une série $\sum_{k\geq 0} H(k)$ est* ***q-hypergéométrique*** *si le rapport entre deux termes consécutifs est une fonction rationnelle de q^k :*
$H(k+1)/H(k) = P(q^k)/Q(q^k)$ où P et Q sont des polynômes en q^k à coefficients dans $\mathbb{C}$.

Définition 7 *Une série hypergéométrique s'écrit sous la forme standard suivante :*

$$
{}_rF_s\left[\begin{matrix}a_1,\ldots,a_r\\ b_1,\ldots,b_s\end{matrix};z\right] := \sum_{k=0}^{\infty} \frac{(a_1)_k \ldots (a_r)_k}{(b_1)_k \ldots (b_s)_k}\,\frac{z^k}{k!}\ , \tag{1.1}
$$

où r et s sont respectivement les nombres de paramètres "haut" et "bas" de la série. Cette série est définie uniquement si $\forall i = 1,\ldots,s, \quad b_i \notin \{0,-1,-2,\ldots\}$.

La manipulation de ces séries est plus aisée sous cette forme standard (appelée aussi notation hypergéométrique). L'algorithme suivant permet de déterminer à partir d'une série hypergéométrique quelconque, la forme standard associée :

- ETAPE 1 : Etant donnée une série $\sum_k t_k$. Décaler l'indice de sommation k tel que la somme commence à $k=0$ avec un terme non nul. Factoriser par le terme correspondant à $k=0$ tel que le premier terme de la somme soit 1.
- ETAPE 2 : Simplifier le rapport t_{k+1}/t_k sous la forme $P(k)/Q(k)$ où P, Q sont des polynômes. Si cette simplification est impossible, la série n'est pas hypergéométrique.
- ETAPE 3 : Factoriser complètement les polynômes P et Q en facteurs linéaires :

$$
\frac{P(k)}{Q(k)} = \frac{(k+a_1)\ (k+a_2)\ \ldots\ (k+a_r)}{(k+b_1)\ (k+b_2)\ \ldots\ (k+b_s)\ (k+1)}\ z\ .
$$

  Si le facteur $(k+1)$ est absent au dénominateur, il faut l'insérer et compenser au numérateur.
- ETAPE 4 : L'identification est terminée :

$$
{}_rF_s\left[\begin{matrix}a_1,\ldots,a_r\\ b_1,\ldots,b_s\end{matrix};z\right]\ .
$$

Exemple : Considérons la série suivante :

$$
S := \sum_k \frac{1}{(2k+1)\ (2k+3)!}\ .
$$

- La plus petite valeur de k telle que le sommant de la série soit défini et non nul est $k=-1$.

$$
S = \sum_{k\geq -1} \frac{1}{(2k+1)\ (2k+3)!} = \sum_{k\geq 0} \frac{1}{(2k-1)\ (2k+1)!}\ .
$$

  La factorisation (afin d'avoir le premier terme égal à 1) est triviale :

$$
S = -\sum_{k\geq 0} \frac{1}{(1-2k)\ (2k+1)!} = -\sum_k s_k\ .
$$

- Le rapport de deux termes consécutifs est une fonction rationnelle qui se factorise :

$$\begin{aligned}\frac{s_{k+1}}{s_k} &= \frac{(1-2k)\,(2k+1)!}{(-1-2k)\,(2k+3)!}\\ &= \frac{1}{4}\,\frac{(k-\frac{1}{2})}{(k+\frac{1}{2})\,(k+\frac{3}{2})\,(k+1)}\ .\end{aligned}$$

- Finalement :

$$S = \sum_k \frac{1}{(2k+1)\,(2k+3)!} = -\ {}_1F_2\left[\begin{matrix}-\frac{1}{2}\\ \frac{1}{2},\frac{3}{2}\end{matrix};\frac{1}{4}\right]\ .$$

Ce procédé est au cœur des manipulations des séries hypergéométriques puisque les formules usuelles de sommations et de transformations s'appliquent sur les formes standards.

Une règle de transformation est une relation entre deux même séries hypergéométriques avec des paramètres différents, citons, par exemple, les deux formules élémentaires de transformation :

$${}_2F_1\left[\begin{matrix}a,b\\ c\end{matrix};z\right] = (1-z)^{-a}\ {}_2F_1\left[\begin{matrix}a,c-b\\ c\end{matrix};\frac{z}{z-1}\right]\ , \tag{1.2}$$

$${}_2F_1\left[\begin{matrix}a,b\\ c\end{matrix};z\right] = (1-z)^{c-a-b}\ {}_2F_1\left[\begin{matrix}c-a,c-b\\ c\end{matrix};z\right]\ . \tag{1.3}$$

Il est possible de répertorier de nombreuses règles de transformation. Le package `HYP` (c.f. Annexes A.8) permet, ainsi, l'application directe de plusieurs formules de transformation.

1.2.2 Les séries hypergéométriques basiques

Enfin, les q-séries (c.f. définition 6 page 5) possèdent également une forme standard facilitant leur manipulation :

Définition 8 *Une série q-hypergéométrique (ou série* ***hypergéométrique basique****) s'écrit sous la forme standard suivante :*

$${}_r\phi_s\left[\begin{matrix}a_1,\ldots,a_r\\ b_1,\ldots,b_s\end{matrix};q,z\right] := \sum_{k=0}^{\infty}\frac{(a_1,\ldots,a_r;q)_k}{(b_1,\ldots,b_s,q;q)_k}\,\left((-1)^k\,q^{k(k-1)/2}\right)^{s-r+1}\ z^k\ , \tag{1.4}$$

où r et s sont respectivement les nombres de paramètres "haut" et "bas", et q la base de la série. Cette série est définie uniquement si $\forall i=1,\ldots,s,\quad b_i \notin \{1,q^{-1},q^{-2},\ldots\}$.

Cette écriture est une conséquence de la généralisation des séries de Heine (c.f. [6], section 1.2). Précisons qu'une autre notation est également employée :

$${}_r\phi_s\left[\begin{matrix}a_1,\ldots,a_r\\ b_1,\ldots,b_s\end{matrix};q,z\right] := \sum_{k=0}^{\infty}\frac{(a_1,\ldots,a_r;q)_k}{(b_1,\ldots,b_s,q;q)_k}\,z^k\ .$$

On vérifie d'après le terme droit de (1.4) que :

$$\frac{(k+1)^{\text{ème}}\ \text{terme}}{k^{\text{ème}}\ \text{terme}} = \frac{(1-a_1\,q^k)\ldots(1-a_r\,q^k)\,(-q^k)^{1+s-r}}{(1-b_1\,q^k)\ldots(1-b_s\,q^k)\,(1-q^{k+1})}\ z\ ,$$

est une fonction rationnelle de q^k, donc le sommant de la série (1.4) est effectivement une fonction q-hypergéométrique (c.f. définition 6).

Exemple : deux séries q-hypergéométriques élémentaires (fonctions q-exponentielles) :

$$e_q(z) := {}_1\phi_0\left[\begin{matrix}0\\ -\end{matrix};q,z\right] = \sum_{k=0}^{\infty} \frac{z^k}{(q;q)_k} = \frac{1}{(z;q)_\infty}\ , \qquad |z|<1\ ,$$

$$E_q(z) := {}_0\phi_0\left[\begin{matrix}-\\ -\end{matrix};q,-z\right] = \sum_{k=0}^{\infty} \frac{q^{k(k-1)/2}\, z^k}{(q;q)_k} = (-z;q)_\infty\ , \qquad z\in\mathbb{C}\ .$$

Remarque : On vérifie rapidement que $(a;q^{-1})_k = (-1)^k\, a^k\, q^{-k(k-1)/2}\, (a^{-1};q)_k$, donc toute série de la forme (1.4) peut être transformée en une série de base q^{-1}. Ainsi, l'étude des q-séries se limite à $|q|\le 1$.

Pour terminer la présentation des q-séries, donnons le q-analogue des formules de transformation (1.2) et (1.3) :

$$ {}_2\phi_1\left[\begin{matrix}a,b\\ c\end{matrix};q,z\right] = \frac{(az;q)_\infty}{(z;q)_\infty}\, {}_2\phi_2\left[\begin{matrix}a,\frac{c}{b}\\ c,az\end{matrix};q,bz\right]\ , \tag{1.5}$$

$$ {}_2\phi_1\left[\begin{matrix}a,b\\ c\end{matrix};q,z\right] = \frac{(abz/c;q)_\infty}{(z;q)_\infty}\, {}_2\phi_1\left[\begin{matrix}\frac{c}{a},\frac{c}{b}\\ c\end{matrix};q,\frac{abz}{c}\right]\ . \tag{1.6}$$

1.3 Forme canonique

Une identité est une équation mathématique qui établit que deux écritures différentes représentent le même objet mathématique. Ainsi, $2+2=4$ est une identité qui n'étonne personne. Par contre, d'autres sont beaucoup moins triviales :

$$\begin{aligned}
(x+1)^2 &= x^2+2x+1\ ,\\
95800^4+217519^4+414560^4 &= 422481^4\ ,\\
1+\sum_{k=1}^{\infty}\frac{x^{m^2}}{(1-x)\,(1-x^2)\,\ldots\,(1-x^m)} &= \prod_{m=0}^{\infty}\frac{1}{(1-x^{2m+1})\,(1-x^{5m+4})}\ ,\\
\sum_k \binom{n}{k} &= 2^n\ ,\\
\sum_{j=0}^{n}\binom{n}{j}^2 &= \binom{2n}{n}\ ,\\
\sum_{i+j+k=n}\binom{i+j}{i}\binom{j+k}{j}\binom{k+i}{k} &= \sum_{0\le j\le n}\binom{2j}{j}\ ,\\
\sum_{s=0}^{2m}(-1)^s\binom{2m}{s}^3 &= (-1)^m\,\frac{(3m)!}{(m!)^3}\ .
\end{aligned}$$

Définition 9 *Une* ***forme canonique*** *d'un objet est une définition sans équivoque de cet objet.*

Exemple : $5 + 7 = 12 = 1 \cdot 10^1 + 2 \cdot 10^0$; une forme canonique qui définit cet entier est sa représentation décimale (ou bien sa représentation binaire). [Remarque : il n'y a pas unicité de la forme canonique.]

Etant donnée une classe d'objets, il existe de nombreuses façons de déterminer un objet particulier. Pour montrer une égalité du type $A = B$, on détermine une forme canonique puis on détermine celle des deux objets A et B (c'est-à-dire $c(A)$ et $c(B)$) puis on vérifie que $c(A) = c(B)$.

Dans ce rapport, on utilisera cette notion : la forme canonique d'un objet donné sera une relation de récurrence satisfaite par cet objet (ainsi que ses conditions initiales).

Chapitre 2

Algorithme de Doron Zeilberger

Dans ce chapitre, nous présentons deux méthodes (c.f. [22]) qui déterminent une relation de récurrence satisfaite par une série. La première méthode utilise le processus d'élimination par l'algorithme d'Euclide. La seconde méthode fait appel à l'algorithme de Gosper [7]. L'objectif final est évidemment de déterminer explicitement la série. On s'intéresse plus particulièrement aux sommes de fonction à deux variables.

2.1 Données du problème

Soit $F(n,k)$, une fonction hypergéométrique en n et k. Les opérateurs de décalage N et K vérifient $N \cdot f(n,k) = f(n+1,k)$ et $K \cdot g(n,k) = g(n,k+1)$. On identifie les opérateurs n et k par la multiplication par n et par k. Les propriétés suivantes sont vérifiées : $Nn = (n+1)N$, $Kk = (k+1)K$ et également $NK = KN$, $Nk = kN$, $Kn = nK$ et $nk = kn$.

On veut trouver une relation de récurrence satisfaite par $a(n) := \sum_k F(n,k)$:

$$s_0(n)\, a(n) + s_1(n)\, a(n+1) + \cdots + s_L(n)\, a(n+L) = 0 \ , \tag{2.1}$$

où les s_i sont des polynômes en n.

En posant $S(N,n) = \sum_{i=0}^{L} s_i(n)\, N^i$, la récurrence s'écrit donc :

$$S(N,n)\, a(n) = 0 \ .$$

On cherche à déterminer un tel $S(N,n)$.

La méthode de *"Creative Telescoping"* consiste à construire une autre fonction $G(n,k)$ qui vérifie :

$$G(n,k+1) - G(n,k) = S(N,n)\, F(n,k) \ . \tag{2.2}$$

Si on trouve une telle fonction $G(n,k)$, on a déterminé (2.1) en sommant les deux membres de l'égalité (2.2) sur k.

2.2 Exemple résolu

$$\text{Soit } F(n,k) = \binom{n}{k}\binom{b}{k} .$$

Le principe mentionné ci-dessus et les méthodes développées ensuite permettent de trouver la fonction :

$$G(n,k) = -N\,k^2\,F(n,k) = \frac{-(n+1)!\,b!}{(k-1)!^2\,(n-k+1)!\,(b-k)!} ,$$

qui satisfait la relation suivante :

$$G(n,k+1) - G(n,k) = S(N,n)\,F(n,k) ,$$

avec $S(N,n) = (n+1)\,[(n+1)\,N - (n+b+1)]$.

Finalement :

$$(n+1)\,a(n+1) - (n+b+1)\,a(n) = 0 ,$$

où $a(n) = \sum_k \binom{n}{k}\binom{b}{k}$.

2.3 Elimination par l'algorithme d'Euclide

2.3.1 Présentation de la méthode

Cette première méthode est très exigeante en temps de calcul et surtout en mémoire. Un algorithme beaucoup plus performant, utilisant le procédé de sommation de Gosper, sera décrit en 2.4.

Les rapports $F(n+1,k)/F(n,k)$ et $F(n,k+1)/F(n,k)$ déterminent chacun une récurrence sur la fonction F. En effet :

$$\frac{F(n+1,k)}{F(n,k)} = \frac{u(n,k)}{v(n,k)} .$$

On a $v(n,k)\,N\,F(n,k) = u(n,k)\,F(n,k)$, ce qui nous donne :

$$[v(n,k)\,N - u(n)]\,F(n,k) = 0 .$$

On obtient donc deux opérateurs $P(N,K,n,k)$ et $Q(N,K,n,k)$ tels que :

$$\begin{cases} P(N,K,n,k)\,F(n,k) = 0 , \\ \text{et } Q(N,K,n,k)\,F(n,k) = 0 . \end{cases}$$

[On remarquera que P ne dépend pas de K et que Q ne dépend pas de N.]

Soient $A(N,K,n,k)$ et $B(N,K,n,k)$ deux polynômes quelconques en N, K, n et k. Les polynômes U tels que $U(N,K,n,k)\,F(n,k) = 0$ forment un idéal à gauche. On peut ainsi écrire :

$$[A(N,K,n,k)\,P(N,K,n,k) + B(N,K,n,k)\,Q(N,K,n,k)]\,F(n,k) = 0 .$$

Il est possible d'éliminer k entre P et Q, c'est-à-dire de trouver $A(N,K,n,k) \neq 0$ et $B(N,K,n,k) \neq 0$ tels que :

$$R(N,K,n) = A(N,K,n,k)\ P(N,K,n,k) + B(N,K,n,k)\ Q(N,K,n,k) \ .$$

Ensuite, on effectue la division euclidienne de R par $K-1$:

$$R(N,K,n) = R(N,1,n) + [(K-1)\ \overline{R}(N,K,n)] \ ,$$

que l'on note encore :

$$R(N,K,n) = S(N,n) + [(K-1)\ \overline{R}(N,K,n)] \ .$$

Or $R(N,K,n)\ F(n,k) = 0$, donc :

$$S(N,n)\ F(n,k) + (K-1)\ \overline{R}(N,K,n)\ F(n,k) = 0 \ ,$$

soit

$$S(N,n)\ F(n,k) = (K-1)\ [-\overline{R}(N,K,n)\ F(n,k)] \ .$$

En posant :

$$G(n,k) = -\overline{R}(N,K,n)\ F(n,k) \ , \tag{2.3}$$

on obtient la relation souhaitée :

$$S(N,n)\ F(n,k) = G(n,k+1) - G(n,k) \ .$$

En sommant sur k :

$$S(N,n)\ \sum_k F(n,k) = \sum_k (G(n,k+1) - G(n,k)) \ .$$

Si F vérifie la condition supplémentaire $F(n,\pm\infty) = 0$ pour tout n, alors après passage à la limite sur k dans l'expression (2.3), on obtient $G(n,\pm\infty) = 0$.

Finalement, la somme de droite se "téléscope" :

$$S(N,n)\ \sum_k F(n,k) = S(N,n)\ a(n) = 0 \ .$$

2.3.2 Exemple

De nombreuses identités font intervenir les coefficients binomiaux. Par conséquent, il est intéressant de les étudier. Appliquons la méthode développée ci-dessus :

$$F(n,k) = \frac{n!}{k!\ (n-k)!} \ , \tag{2.4}$$

$$\begin{cases} F(n+1,k)/F(n,k) = (n+1)/(n-k+1) \ , \\ F(n,k+1)/F(n,k) = (n-k)/(k+1) \ . \end{cases}$$

Le produit en croix donne :

$$\begin{cases} (n-k+1)\ F(n+1,k) - (n+1)\ F(n,k) = 0 \ , \\ (k+1)\ F(n,k+1) - (n-k)\ F(n,k) = 0 \ . \end{cases}$$

En notation compacte :

$$\begin{cases} [(n-k+1)\,N-(n+1)]\,F(n,k)=0\ , & \textit{(i)} \\ [(k+1)\,K-(n-k)]\,F(n,k)=0\ . & \textit{(ii)} \end{cases}$$

On élimine k :

$$(K+1)\ \textit{(i)}+N\ \textit{(ii)}=\Big[(K+1)\,[(n+1)\,N-(n+1)]+N\,(-n)\Big]\,F(n,k)=0\ ,$$

$$(n+1)\,[NK-K-1]\,F=0\ ,$$

$$R(N,K,n)=(n+1)\,[NK-K-1]\ .$$

Finalement :

$$\begin{aligned} S(N,n) &= R(N,1,n) \\ &= (n+1)\,(N-2)\ , \end{aligned}$$

$$\Longrightarrow (n+1)\,[a(n+1)-2\,a(n)]=0\ .$$

La valeur initiale est :

$$a(0)=1\ .$$

Conclusion :

$$a(n)=2^n\ .$$

2.4 Méthode rapide de Doron Zeilberger

Dans cette section, nous présentons tout d'abord l'algorithme de Gosper [7], puis l'utilisation de cet algorithme pour démontrer des identités et également pour trouver la récurrence satisfaite par une série.

2.4.1 Algorithme de Gosper

Description de l'algorithme

Soit une suite hypergéometrique $b(n)$. L'algorithme de Gosper est une procédure de décision qui permet de trouver (si elle existe) une suite hypergéométrique $T(n)$ vérifiant $T(n)-T(n-1)=b(n)$. La méthode se décompose de la façon suivante :

- Cherchons tout d'abord la décomposition suivante de $\frac{b(n)}{b(n-1)}$:

$$\frac{b(n)}{b(n-1)}=\frac{p(n)}{p(n-1)}\,\frac{q(n)}{r(n)}\ , \tag{2.5}$$

avec $p(n)$ maximal, c'est-à-dire tel que $q(n)$ et $r(n)$ vérifient $\mathrm{pgcd}(q(k),r(k+j))=1$ pour tout $j\geq 0$,

où $p(n)/p(n-1)$ est appelée *"partie polynomiale"* tandis que $q(n)/r(n)$ est dite *"partie factorielle pure"*.

Pour cela, on écrit la décomposition initiale avec $p(n) = 1$; ensuite, s'il existe un $j \geq 0$ tel que $\mathrm{pgcd}(q(n), r(n+j)) = g(n) \neq 1$, alors on pose :

$$\begin{aligned} q'(n) &= q(n)/g(n) \ , \\ r'(n) &= r(n)/g(n-j) \ , \\ p'(n) &= p(n)\, g(n)\, g(n-1) \cdots g(n-j+1) \ , \end{aligned}$$

et la décomposition de $b(n)/b(n-1)$ devient :

$$\frac{b(n)}{b(n-1)} = \frac{p'(n)}{p'(n-1)} \frac{q'(n)}{r'(n)} \ .$$

En itérant ce processus, Gosper [7] montre que l'on obtient une décomposition de la forme (2.5) avec la condition souhaitée ($\mathrm{pgcd}(q(n), r(n+j)) = 1$ pour tout $j \geq 0$).

- Déterminons maintenant $T(n)$ de la forme :

$$T(n) = \frac{b(n)\, q(n+1)}{p(n)} f(n) \ ,$$

où $p(n)$ et $q(n)$ sont les polynômes précédemment définis.

- Précisons la nature de $f(n)$:

$$\frac{T(n)}{b(n)} = \frac{T(n)}{T(n) - T(n-1)} = \frac{t(n)}{t(n)-1} \ ,$$

où $t(n) = T(n)/T(n-1)$ est une fonction rationnelle de n. Donc $T(n)/b(n)$ est une fonction rationnelle de n.

Or :

$$\frac{T(n)}{b(n)} = \frac{q(n+1)}{p(n)} f(n) \ ,$$

donc $f(n)$ est également une fonction rationnelle de n.

Supposons que $f(n) = x(n)/y(n)$ avec $y(n) \neq 1$ où $x(n)$ et $y(n)$ sont des polynômes premiers entre eux. Soit j le plus grand entier tel que $\mathrm{pgcd}(y(n), y(n+j)) \neq 1$. Alors, on arrive à une contradiction à partir de l'équation fonctionnelle (2.6), des hypothèses sur $q(n)$ et $r(n)$ et de la maximalité de j (c.f. [7]). Donc $f(n)$ est un polynôme.

- Il reste à déterminer le polynôme $f(n)$. Pour cela, on écrit :

$$T(n) - T(n-1) = b(n) \ ,$$

$$\frac{b(n)\, q(n+1)}{p(n)} f(n) - \frac{b(n-1)\, q(n)}{p(n-1)} f(n-1) = b(n) \ ,$$

$$\text{par (2.5) :} \qquad \frac{b(n)\, q(n+1)}{p(n)} f(n) - \frac{b(n)\, r(n)}{p(n)} f(n-1) = b(n) \ ,$$

$$q(n+1)\, f(n) - r(n)\, f(n-1) = p(n) \ . \tag{2.6}$$

$\Rightarrow$ Si cette équation fonctionnelle admet une solution polynomiale pour f, le problème est réglé : T est complètement déterminé. Sinon, la suite $T(n)$ cherchée n'existe pas.

Exemple

On s'intéresse à :

$$b(n) = (4n+1)\,\frac{n!}{(2n+1)!}\ .$$

- Cherchons tout d'abord la décomposition de $\frac{b(n)}{b(n-1)}$:

$$\frac{b(n)}{b(n-1)} = \frac{(4n+1)\frac{n!}{(2n+1)!}}{(4n-3)\frac{(n-1)!}{(2n-1)!}} = \frac{(4n+1)}{2(4n-3)(2n+1)}\ , \tag{2.7}$$

  avec $p(n) = 4n+1$, $q(n) = 1$ et $r(n) = 2(2n+1)$.

- $T(n)$ s'écrit :

$$T(n) = \frac{4n+1\ \frac{n!}{(2n+1)!}}{4n+1}\,f(n) = \frac{n!}{(2n+1)!}\,f(n)\ .$$

- L'équation fonctionnelle (2.6) s'écrit :

$$f(n) - 2(2n+1)\,f(n-1) = 4n+1\ , \tag{2.8}$$

  $\Longrightarrow f(n) = -1$.

- Finalement :

$$T(n) = -\frac{n!}{(2n+1)!}\ .$$

2.4.2 Méthode de Wilf-Zeilberger

Cette méthode est une application directe de l'algorithme de Gosper pour démontrer des identités de la forme :

$$a(n) := \sum_k F(n,k) = \texttt{const}\ .$$

[Dans le cas de $\sum_k F(n,k) = r(n)$ où $r(n) \neq 0$, on peut se ramener à ce type d'équation en posant $F(n,k) := F(n,k)/r(n)$.]

L'idée est d'appliquer l'algorithme de Gosper à l'expression $b_k = F(n,k) - F(n-1,k)$ par rapport à la variable k. On obtient alors (si elle existe) une forme $G(n,k)$ telle que :

$$b_k = F(n,k) - F(n-1,k) = G(n,k) - G(n,k-1)\ . \tag{2.9}$$

Il y a ici, une complète symmétrie entre les indices n et k : (F,G) est apellée paire **WZ**.

En sommant (2.9) sur k, on obtient :

$$\sum_k \left(F(n,k) - F(n-1,k)\right) = \sum_k \left(G(n,k) - G(n,k-1)\right)\ .$$

Le terme de droite se "téléscope" et le terme de gauche s'écrit $a(n) - a(n-1)$. D'où :

$$a(n) - a(n-1) = 0\ ,$$

$$a(n) = a(n-1) = \texttt{const}\ . \tag{2.10}$$

Bonus : "Companion Identity"

Une paire WZ permet d'obtenir une égalité de la forme (2.10), mais aussi une autre égalité en sommant (2.9) sur n :

$$\sum_n (F(n,k) - F(n-1,k)) = \sum_n (G(n,k) - G(n,k-1)) \ .$$

Si F vérifie la condition $F(\pm\infty, k) = 0$ pour tout k, alors le terme gauche se téléscope, et l'on obtient :

$$\sum_n G(n,k) = \texttt{const} \ .$$

Cette dernière égalité est une nouvelle identité, au sens où elle ne se déduit pas directement de (2.10). On la nomme *"Companion Identity"*.

Identités duales

Le passage entre une identité et sa "Companion Identity" n'est pas une involution. Cependant, il existe une vraie identité duale dans la théorie de Wilf et Zeilberger.
Soit (F, G) une paire WZ.
Supposons que le sommant $F(n,k)$ s'écrit sous la forme :

$$F(n,k) = x^n \, y^k \, \rho(n,k) \, \frac{\prod_i (a_i \, n + b_i \, k + c_i)!}{\prod_i (u_i \, n + v_i \, k + w_i)!} \ ,$$

où ρ est une fonction rationnelle en n et k.
Effectuons l'opération suivante sur la fonction F :

- Cherchons les facteurs de la forme $(a\,n + b\,k + c)!$,
- Remplaçons ces facteurs par $(-1)^{an+bk}/(-a\,n - b\,k - c - 1)!$ (excepté ceux de la forme $(a\,n - a\,k + c)!$ que l'on laisse inchangés).

Ainsi, nous avons transformé F en $\widetilde{F}$ et en répétant cette opération sur G, on obtient $\widetilde{G}$. $(\widetilde{F}, \widetilde{G})$ est alors une autre paire WZ[1].
Exemple : soit la paire WZ suivante :

$$F(n,k) = \frac{1}{2^{n+1}} \binom{n+1}{k} \ ; \qquad G(n,k) = \frac{-1}{2^{n+1}} \binom{n}{k} \ . \tag{2.11}$$

La transformation décrite ci-dessus permet d'obtenir la paire duale suivante :

$$\widetilde{F}(n,k) = \frac{(-1)^{n+k}}{2^{n+1}} \binom{-k-1}{-n-2} \ ; \qquad \widetilde{G}(n,k) = \frac{(-1)^{n+k}}{2^{n+1}} \binom{-k-1}{-n-1} \ . \tag{2.12}$$

En appliquant une seconde fois cette transformation, on retrouve la paire initiale (2.11).
En sommant (2.9) sur k (avec la paire (2.11)), on obtient l'égalité suivante :

$$\sum_k \frac{1}{2^n} \binom{n}{k} = 1 \ .$$

[1]Ce résultat est du à Wilf et Zeilberger [18].

L'identité duale se déduit en sommant (2.9) sur n (avec la paire (2.12)) :

$$\sum_n (-2)^n \binom{k}{n} = (-1)^k \ .$$

Ce processus permet de découvrir de nouvelles identités. De nombreuses égalités ont des paramètres libres, la spécialisation de ces paramètres nous donne encore de nouvelles identités.

Le procédé de dualisation ne commute pas, en général, avec la spécialisation. Ainsi, on peut imaginer commencer avec une identité donnée, la spécialiser, dualiser, effectuer une seconde spécialisation, etc ..., on génère ainsi une suite d'identités.

2.4.3 Algorithme rapide de Zeilberger

Revenons au problème (plus général) posé en 2.1. On s'intéresse maintenant à un algorithme plus performant qui utilise l'algorithme de Gosper pour déterminer une forme close $G(n,k)$ telle que :

$$S(N,n)\, F(n,k) = G(n,k+1) - G(n,k) \ . \tag{2.13}$$

Le problème est que nous ne connaissons pas $S(N,n)$.

Une version modifiée de l'algorithme de Gosper permet de déterminer simultanément les inconnues du problème $S(N,n)$ et $G(n,k)$.

- Tout d'abord, on écrit $S(N,n)$ sous forme générique :

  $$S(N,n) = \sum_{i=0}^{I} s_i(n)\, N^i \ ,$$

  où I est l'ordre de la récurrence, les coefficients $s_i(n)$ étant des polynômes en n.

- Alors :

  $$H(n,k) = S(N,n)\, F(n,k) = \sum_{i=0}^{I} s_i(n)\, F(n+i,k) \ ,$$

  $$H(n,k) = \left[\sum_{i=0}^{I} s_i(n)\, \frac{F(n+i,k)}{F(n,k)}\right] F(n,k) \ .$$

  L'expression entre les crochets est une fonction rationnelle en n et k dont le numérateur est une expression linéaire de $s_0(n), s_1(n), \ldots, s_I(n)$.

- On cherche alors la décomposition (2.5) de $H(n,k)$ par rapport à la variable k en appliquant l'algorithme de Gosper :

  $$\frac{H(n,k)}{H(n,k-1)} = \frac{p(k)}{p(k-1)}\, \frac{q(k)}{r(k)} \ .$$

  L'équation fonctionnelle (2.6) est de la forme :

  $$p(k) = q(k+1)\, f(k) - r(k)\, f(k-1) \ ,$$

  où $p(k)$ est une expression linéaire des s_i.

On cherche $f(k)$ exprimée sous la forme générique $f(k) = f_0 + f_1 k + \cdots + f_L k^L$, en comparant les coefficients des puissances de k.

On obtient ainsi un système d'équations linéaires en $f_0, f_1, \ldots, f_L, s_0, s_1, \ldots, s_I$.

Si ce système admet une solution, on a alors :

$$G(n,k) = \frac{H(n,k)}{p(k)}\, q(k+1)\, f(k) \ ,$$

$$\text{et } S(N,n) = \sum_{i=0}^{I} s_i(n)\, N^i \ .$$

Sinon (si le système n'admet pas de solution), cela signifie qu'il n'existe pas de telle récurrence d'ordre I, et on peut recommencer en remplaçant I par $I+1$.

Si $F(n,k)$ est une fonction *holonome*[2], alors la théorie de l'holonomie (c.f. [21]) implique que $\sum_k F(n,k)$ vérifie toujours une récurrence linéaire à coefficients polynomiaux. Avec cette hypothèse supplémentaire, l'algorithme termine.

Finalement, après avoir déterminé $G(n,k)$, la sommation sur k de l'égalité (2.13) donne l'égalité recherchée :

$$S(N,n)\, a(n) = 0 \ .$$

2.4.4 Exemple

$$F(n,k) = \frac{1}{k!\,(n-k)!} \ . \tag{2.14}$$

- Tentative avec $I = 0$: Échec.
- $I = 1$:

$$H(n,k) = S(N,n)\, F(n,k) = s_0(n)\, F(n,k) + s_1(n)\, F(n+1,k) \ ,$$

$$H(n,k) = \frac{s_0(n)}{k!\,(n-k)!} + \frac{s_1(n)}{k!\,(n+1-k)!} \ ,$$

$$H(n,k) = \frac{s_0(n)\,(n+1-k) + s_1(n)}{k!\,(n+1-k)!} \ .$$

Les polynômes p, q, r, s'écrivent initialement par l'algorithme de Gosper :

$$p(k) = (n+1-k)\, s_0(n) + s_1(n) \ ,$$

$$\frac{q(k)}{r(k)} = \frac{(k-1)!\,(n-k+2)!}{k!\,(n+1-k)!} = \frac{(n+2-k)}{k} \ .$$

Précisément, $q(k) = n+2-k$ et $r(k) = k$. On vérifie que $\text{pgcd}(q(k), r(k+j)) = 1$ pour tout $j \geq 0$.

[2] c.f. définition des suites et des fonctions holonomes page 33.

Ainsi, les polynômes s'écrivent :

$$\begin{aligned} p(k) &= (n+1-k)\; s_0(n) + s_1(n) \;, \\ q(k) &= n+2-k \;, \\ r(k) &= k \;. \end{aligned}$$

L'équation fonctionnelle (2.6) s'écrit :

$$(n-k+1)\; f(k) - k\, f(k-1) = (n+1-k)\; s_0(n) + s_1(n) \;.$$

On en déduit le degré en k de f : $1 + deg_k\ f = 1 \Rightarrow deg_k\ f = 0$. On pose $f(k) = f_0$. En comparant les coefficients de k^0 et k^1, le système à résoudre devient :

$$\begin{aligned} (n+1)\; f_0 &= (n+1)\; s_0(n) + s_1(n) \;, \\ -2\, f_0 &= -s_0(n) \;. \end{aligned}$$

En normalisant $f_0 = f = 1$, on obtient la récurrence :

$$\begin{aligned} s_0 &= 2 \;, \\ s_1 &= -(n+1) \;, \end{aligned}$$

qui s'écrit finalement :

$$2\, a(n) - (n+1)\; a(n+1) = 0 \;.$$

Ceci joint à la condition initiale $a(0) = 1$ donne la solution :

$$a(n) = \sum_{k=0}^{n} \frac{1}{k!\;(n-k)!} = \frac{2^n}{n!} \;.$$

2.5 Résultats : cas particuliers

Les méthodes présentées permettent de déterminer une relation de récurrence satisfaite par une série hypergéométrique :

$$\sum_{i=0}^{L} s_i(n)\; a(n+i) = 0 \;.$$

- Si $L = 1$, la récurrence $s_0(n)\; a(n) + s_1(n)\; a(n+1) = 0$ est du premier ordre. Dans ce cas, $a(n+1)/a(n) = -s_0(n)/s_1(n)$ est une fonction rationnelle en n. On a ainsi une formule explicite : $a(n) = a(0)\ \prod_{j=0}^{n-1}(-s_0(j)/s_1(j))$.
- Si $L > 1$ et si les coefficients $s_i(n)$ sont constants, alors on sait résoudre ce type de récurrence à coefficients constants : une fois encore, on peut obtenir une formule explicite pour la série.
- Sinon, on ne peut pas définir complètement la série, mais on dispose alors d'informations sur son comportement asymptotique et l'on dispose également d'éléments pour démontrer des identités faisant intervenir cette série.

2.6 Application

□ Les nombres d'*Apéry* A_n et de *Franel* F_n définis ci-dessous :

$$A_n = \sum_{k=0}^{n} \binom{n}{k}^2 \binom{n+k}{k}^2 ,$$

$$F_n = \sum_{k=0}^{n} \binom{n}{k}^3 ,$$

vérifient l'identité suivante :

$$\sum_{k=0}^{n} \binom{n}{k}^2 \binom{n+k}{k}^2 = \sum_{k=0}^{n} \binom{n}{k} \binom{n+k}{k} \sum_{j=0}^{k} \binom{k}{j}^3 . \tag{2.15}$$

□ $p(n,k) = \binom{n}{k}^3$ est hypergéométrique :

$$\begin{cases} [(n+1-k)^3\, N - (n+1)^3]\, p(n,k) = 0 \ , \\ [(n-k)^3\, K - (k+1)^3]\, p(n,k) = 0 \ . \end{cases}$$

Le *"creative telescoping"* donne l'opérateur P_n :

$$P_n \ : \ (n+2)^2\, N^2 - (7n^2 + 21n + 16)\, N - 8(n+1)^2 \ .$$

$q(n,k) = \binom{n}{k} \binom{n+k}{k}$ est hypergéométrique :

$$\begin{cases} [(n+1-k)\, N - (n+1+k)]\, q(n,k) = 0 \ , \\ [(k+1)^2\, K - (n(n+1) - k(k+1))]\, q(n,k) = 0 \ . \end{cases}$$

Le produit avec P_n donne :

$$\begin{cases} [(n+1-k)\, N - (n+1+k)]\, q(n,k) = 0 \ , \\ [(k+2)^4\, K^2 + 8(n+k+2)(n+k+1)(n-k) \\ \qquad -(7k^2 + 21k + 16)(n+k+2)(n-k-1)\, K]\, q(n,k) = 0 \ . \end{cases}$$

Le *"creative telescoping"* donne la récurrence satisfaite par le membre droit B_n de l'équation (2.15) :

$$(n+2)^3\, N^2 - (2n+3)(17n^2 + 51n + 39)\, N + (n+1)^3 \ .$$

□ $r(n,k) = \binom{n}{k}^2 \binom{n+k}{k}^2$ est hypergéométrique :

$$\begin{cases} [(n+1-k)^2\, N - (n+1+k)^2]\, r(n,k) = 0 \ , \\ [(k+1)^4\, K - (n(n+1) - k(k+1))^2]\, r(n,k) = 0 \ . \end{cases}$$

Le *"creative telescoping"* donne la récurrence satisfaite par les nombres d'*Apéry* A_n :

$$(n+2)^3\,N^2 - (2n+3)(17n^2+51n+39)\,N + (n+1)^3\ .$$

$\Rightarrow$ Les deux membres de l'égalité vérifient la même récurrence.

$\square$ La comparaison des valeurs initiales $A_0 = B_0 = 1$ et $A_1 = B_1 = 5$ terminent la preuve de l'équation (2.15).

2.7 Implémentations

L'algorithme de Zeilberger a été implémenté en **MAPLE** par Doron Zeilberger lui-même, par Tom H. Koornwinder [9] et aussi en **MATHEMATICA** par Peter Paule et Markus Schorn [12].

2.7.1 Package Ekhad de Doron Zeilberger

```
$ maple
    |\^/|      Maple V Release 3 (Universite de Marne La Vallee)
._|\|   |/|_.  Copyright (c) 1981-1994 by Waterloo Maple Software and the
 \  MAPLE  /   University of Waterloo. All rights reserved. Maple and Maple V
 <____ ____>   are registered trademarks of Waterloo Maple Software.
      |        Type ? for help.
> read EKHAD;

                             Version of July 16, 1996


       This version corrects a subtle bug discovered by Helmut Prodinger


           Previous versions benefited from comments by Paula Cohen,

                         Lyle Ramshaw, and Bob Sulanke.


                    This is EKHAD, One of the Maple packages

                             accompanying the book

                                    "A=B"

                   (published by A.K. Peters, Welesley, 1996)

              by Marko Petkovsek, Herb Wilf, and Doron Zeilberger.


                  The most current version is available on WWW at:

                      http://www.math.temple.edu/~zeilberg .

        Information about the book, and how to order it, can be found in

               http://www.central.cis.upenn.edu/~wilf/AeqB.html .
```

```
          Please report all bugs to: zeilberg@math.temple.edu .

      All bugs or other comments used will be acknowledged in future

                              versions.

          For general help, and a list of the available functions,

      type "ezra();". For specific help type "ezra(procedure_name)"

> zeil(binomial(n,k),k,n,N);

                                         k
                              2 - N, ---------
                                     n - k + 1


> zeil(binomial(n,k)^2,k,n,N);

                                                       2
                                         (3 n + 3 - 2 k) k
                  - 4 n - 2 + (n + 1) N, - -----------------
                                                     2
                                            (n - k + 1)
```

Dans les exemples ci-dessus, la fonction retourne deux expressions qui correspondent respectivement à la récurrence S et à la fonction R tel que le sommant F passé en argument vérifie l'équation $S(N,n)\,F(n,k) = G(n,k+1) - G(n,k)$ où $G(n,k) := R(n,k)\,F(n,k)$.

2.7.2 Package `Zeilb` de Tom H. Koornwinder

Ce package travaille directement sur les formes standards des séries hypergéométriques (c.f. définition 7 page 6). Ici, la récurrence satisfaite par la série ${}_2F_1\left[{-n,b \atop c};z\right]$ est d'ordre 2 :

```
> read zeilb;
> zeilb([-n,b],[c],z,n,1);

                             Algorithm fails

> zeilb([-n,b],[c],z,n,2);

    (z n - 2 n + z b - c + 2 - z) INH(n - 1)   (z - 1) (n - 1) INH(n - 2)
  - ---------------------------------------- + -------------------------, 1 < n
                    c - 1 + n                          c - 1 + n
```

2.7.3 Package `Zeil` de Peter Paule et Markus Schorn

Nous présentons ici, deux applications non triviales qui illustrent les techniques étudiées.

Application 1 : Sommation double

$\forall m, n \in \mathbb{N}$, on considère :

$$\sum_{j,k} (-1)^{j+k} \binom{j+k}{j} \binom{r}{j} \binom{n}{k} \binom{m+n-j-k}{m-j} = \binom{n+r}{n} \binom{m-r}{m-n} , \tag{2.16}$$

On applique la formule d'inversion élémentaire :

$$g(r) = \sum_j (-1)^j \binom{r}{j} f(j) \Longleftrightarrow f(r) = \sum_j (-1)^j \binom{r}{j} g(j) ,$$

$$\sum_k (-1)^k \binom{n}{k} \binom{r+k}{r} \binom{m+n-r-k}{m-r} = \sum_j (-1)^j \binom{r}{j} \binom{n+j}{n} \binom{m-j}{m-n} . \tag{2.17}$$

Après avoir vérifié les valeurs initiales, l'algorithme de Zeilberger génère les récurrences satisfaites par les deux membres de l'égalité (2.17) :

```
In[1]:= <<zb_alg.m
Fast Zeilberger by Peter Paule and Markus Schorn. (V 2.2)
Systembreaker = ENullspace

In[2]:= Zb[(-1)^k * Binomial[n,k] * Binomial[k+r,k] * Binomial[m+n-r-k,m-r], k, n, 2]

Out[2]= {(m - n) (2 + m + n) F[k, n] + (3 + 2 n) (-m + 2 r) F[k, 1 + n]

                 2
     + (2 + n)  F[k, 2 + n] == Delta[k, R F[k, n]]}

In[3]:= Zb[(-1)^j * Binomial[r,j] * Binomial[n+j,n] * Binomial[m-j,m-n], j, n, 2]

Out[3]= {(m - n) (2 + m + n) F[j, n] + (3 + 2 n) (-m + 2 r) F[j, 1 + n]

                 2
     + (2 + n)  F[j, 2 + n] == Delta[j, R F[j, n]]}
```

Ces deux récurrences sont identiques, l'égalité (2.16) est alors démontrée.

Application 2 : Etude de la récurrence obtenue par l'algorithme de Zeilberger

L'algorithme de Zeilberger détermine le plus petit L tel qu'il existe $s_0(n), s_1(n), \ldots, s_L(n)$ et $G(n,k)$ qui vérifie l'égalité :

$$s_0(n)\, F(n,k) + s_1(n)\, F(n+1,k) + \cdots + s_L(n)\, F(n+L,k) = G(n,k+1) - G(n,k) .$$

La sommation sur k implique que $a(n) := \sum_k F(n,k)$ satisfait la récurrence :

$$s_0(n)\, a(n) + s_1(n)\, a(n+1) + \cdots + s_L(n)\, a(n+L) = 0 .$$

A priori, il n'y a aucune raison de s'attendre à ce que L soit minimal pour cette récurrence. En étudiant un grand nombre d'identités, Peter Paule et Markus Schorn ont tenté de montrer que c'était le cas. L'exemple suivant leur a prouvé le contraire :

```
In[4]:= Zb[(-1)^k Binomial[n,k] Binomial[3k,n], k,n,1]

Out[4]= Try higher order

In[5]:= Zb[(-1)^k Binomial[n,k] Binomial[3k,n], k,n,2]

Out[5]= {9 (1 + n) (2 + n) F[k, n] + 3 (2 + n) (7 + 5 n) F[k, 1 + n]

    + 2 (2 + n) (3 + 2 n) F[k, 2 + n] == Delta[k, R F[k, n]]}
```

En effet, la récurrence $9(n+1)(n+2)+3(n+2)(5n+7)N+2(n+2)(2n+3)N^2$ est identique à $9(n+1)+3(5n+7)N+2(2n+3)N^2$ mais aussi à $(3(n+1)+2(2n+3)N)\,(N+3)$. L'algorithme de Zeilberger fournit donc une récurrence d'ordre 2, or la série étudiée vérifie une autre récurrence $(N+3)$ d'ordre 1.

Dans le cas général de la série :

$$S_d(n) = \sum_{k=0}^{n} (-1)^k \binom{n}{k} \binom{d\,k}{n} \qquad d > 0 \ ,$$

l'algorithme de Zeilberger fournit une récurrence d'ordre $d-1$ alors que la récurrence minimale est d'ordre 1 et donne la forme $S_d(n) = (-d)^n$ (c.f. [12], section 4.3).

En pratique, les récurrences données par l'algorithme de Zeilberger sont néanmoins très souvent d'ordre minimale.

2.8 Généralisation au cas des fonctions à variables multiples

Soit $F(n, k_1, k_2, \ldots, k_r)$ une fonction hypergéométrique en $n, k_1, k_2, \ldots, k_r$. Le théorème fondamental ([19] section 4.2) garantit qu'il existe des polynômes $s_0(n), s_1(n), \ldots, s_L(n)$ et des fonctions $R_j(n, k_1, k_2, \ldots, k_r)$ telles que $G_i = R_i\,F$ qui satisfont l'égalité suivante :

$$\sum_{i=0}^{L} s_i(n)\, F(n+i, k_1, \ldots, k_r) = \sum_{j=1}^{r} \Big[G_j(n, k_1, \ldots, k_j + 1, \ldots, k_r) - G_j(n, k_1, \ldots, k_j, \ldots, k_r) \Big] \ .$$

En sommant sur $k_1, k_2, \ldots,$ et k_r, les sommes de droite se téléscopent. Ainsi, la somme $a(n) := \sum_{k_1,k_2\ldots,k_r} F(n, k_1, k_2, \ldots, k_r)$ vérifie une équation de récurrence à coefficients polynomiaux :

$$\sum_{i=0}^{L} s_i(n) a(n+i) = 0 \ .$$

Remarque : Dans le cas d'une fonction à deux variables, on a vu qu'on obtenait deux identités ("originale" et "companion"). Ici, on a $r+1$ identités : l'"originale" et r bonus.

Chapitre 3

q-analogue de l'algorithme de Zeilberger

Zeilberger a été le premier à observer que le problème de la sommation de fonctions hypergéométriques pouvait être résolu par l'utilisation de l'algorithme de Gosper. Cette approche a été suivie par Koornwinder [9] qui a remarqué que les algorithmes de Gosper et de Zeilberger pouvaient résoudre également le cas des séries q-hypergéométriques. Ce chapitre est la description du q-analogue de l'algorithme de Zeilberger par l'approche de P. Paule et de A. Riese [13].

3.1 q-factorisation factorielle

3.1.1 Définitions de base

Soit $F = \mathbb{K}(q)$. On note ϵ, l'opérateur q-décalage sur $F[x]$, $\epsilon(p(x)) = p(qx)$ pour tout $p \in F[x]$.

Un polynôme $p \in F[x]$ est **unitaire** si le terme de plus haut degré est égal à 1 ($lcf(p) = 1$).

Un polynôme $p \in F[x]$ est **q-unitaire** si $p(0) = 1$.

Le degré polynomial (en x) de $p \in F[x]$ est noté $deg(p)$. Par convention, on définit $deg(0) := -1$.

Un polynôme $p \in F[x]$ a une unique décomposition q-unitaire de la forme :

$$p = z\, x^\alpha\, \widehat{p}\ ,$$

avec $z \in F$, $\alpha \in \mathbb{N}$ et $\widehat{p} \in F[x]$ q-unitaire.

En plus de cette normalisation, on définit le pgcd_q qui est le pgcd de deux polynômes q-unitaires et qui est lui-même q-unitaire. Si $p_1 = z_1 x_1^\alpha \widehat{p_1}$ et $p_2 = z_2 x_2^\alpha \widehat{p_2}$ sont les décompositions q-unitaires de $p_1, p_2 \in F[x]$, on définit :

$$\mathrm{pgcd}_q(p_1, p_2) := \mathrm{pgcd}(x_1^\alpha, x_2^\alpha)\, \mathrm{pgcd}_q(\widehat{p_1}, \widehat{p_2})\ .$$

La k-ième q-factorielle descendante d'un polynôme p est définie par :

$$[p]^{\frac{k}{q}} = \prod_{i=0}^{k-1} \epsilon^{-i} p\ .$$

Par convention $\prod_{i\in\emptyset} p_i = 1$, d'où $[p]^{\frac{0}{q}} = 1$.

La notion de "q-greatest factorial factorization" (qGFF) d'un polynôme est le q-analogue de la forme GFF introduite par Paule [11] (définie avec l'opérateur q-décalage ϵ au lieu du décalage standard $E(p(x)) = p(x+1)$).

Définition 10 *On dit que $\langle p_1, p_2 \dots, p_k\rangle$ est la plus grande q-factorielle factorisation d'un polynôme q-unitaire $p \in F[x]$ si les conditions suivantes sont vérifiées :*
(qGFF1) $p = [p_1]^{\frac{1}{q}} [p_2]^{\frac{2}{q}} \dots [p_k]^{\frac{k}{q}}$,
(qGFF2) Chaque p_i est q-unitaire et $k > 0$ implique $deg(p_k) > 0$,
(qGFF3) $i \leq j \Rightarrow pgcd_q([p_i]^{\frac{i}{q}}, \epsilon p_j) = 1 = pgcd_q([p_i]^{\frac{i}{q}}, \epsilon^{-j} p_j)$.

Théorème 1 *Si $\langle p_1, p_2, \dots, p_k\rangle$ et $\langle q_1, q_2 \dots, q_k\rangle$ sont deux formes qGFF de $p \in F[x]$, alors $k = l$ et $p_i = q_i$ pour tout $i \in 1, 2, \dots, k$.*

3.1.2 Lemme fondamental

Lemme 1 (Lemme fondamental qGFF) *Etant donné un polynôme q-unitaire $p \in F[x]$, de forme qGFF $\langle p_1, p_2, \dots, p_k\rangle$, alors :*

$$pgcd_q(p, \epsilon p) = [p_1]^{\frac{0}{q}} [p_2]^{\frac{1}{q}} \dots [p_k]^{\frac{k-1}{q}} .$$

3.2 Algorithme q-Telescope

Soit une suite $\langle f_k\rangle$ q-hypergéométrique en k sur F. Il existe alors une fonction rationnelle ρ telle que :

$$\forall k \in \mathbb{N},\ \frac{f_{k+1}}{f_k} = \rho(q^k) .$$

Le problème du *q-hypergéométrique telescoping* revient à trouver une solution $\langle g_k\rangle$ q-hypergéométrique de :

$$g_{k+1} - g_k = f_k . \tag{3.1}$$

Supposons qu'une telle solution existe.

Posons $\sigma \in F[x]$ tel que $g_{k+1}/g_k = \sigma(q^k)$ pour tout $k \in \mathbb{N}$. Alors $g_k = \tau(q^k) \cdot f_k$ où $\tau(x) = 1/(\sigma(x) - 1)$.

L'équation (3.1) devient :

$$\tau(q^{k+1}) \cdot f_{k+1} - \tau(q^k) \cdot f_k = f_k ,$$

$$\tau(q^{k+1}) \cdot f_{k+1}/f_k - \tau(q^k) = 1 ,$$

$$\tau(q^{k+1}) \cdot \rho(q^k) - \tau(q^k) = 1 .$$

Avec $\rho = z \cdot x^\alpha \cdot a/b$, $z \in F$, et $a, b \in F[x]$ relativement premier et q-unitaire, on obtient :

$$z \cdot\ x^\alpha \cdot a/b \cdot \epsilon(\tau) - \tau = 1 .$$

Et en posant $\alpha_+ := \max(\alpha, 0)$ et $\alpha_- := \max(-\alpha, 0)$:

$$z \cdot \ x^{\alpha+\alpha_-} \cdot a \cdot \epsilon(\tau) - x^{\alpha_-} \cdot b \cdot \tau = x^{\alpha_-} \cdot b \ ,$$

$$z \cdot \ x^{\alpha_+} \cdot a \cdot \epsilon(\tau) - x^{\alpha_-} \cdot b \cdot \tau = x^{\alpha_-} \cdot b \ . \tag{3.2}$$

Une solution rationnelle τ de (3.2) donne une solution q-hypergéométrique de (3.1). Cela signifie que le q-hypergéométrique *telescoping* est équivalent à chercher une solution rationnelle de (3.2).

Pour cela, on écrit τ sous la forme $\tau = u/v$ où $u, v \in F[x]$ et $v = x^\beta\, \widehat{v}$ (décomposition q-unitaire de v).

Supposons qu'une telle solution τ existe, et que l'on connaisse v ou un multiple $V \in F[x]$ de v. Alors :

$$z \cdot \ x^{\alpha_+} \cdot a \cdot \frac{\epsilon U}{\epsilon V} - x^{\alpha_-} \cdot b \cdot \frac{U}{V} = x^{\alpha_-} \cdot b \ ,$$

$$z \cdot \ x^{\alpha_+} \cdot a \cdot V \cdot \epsilon U - x^{\alpha_-} \cdot b \cdot U \cdot \epsilon V = x^{\alpha_-} \cdot b \cdot V \cdot \epsilon V \ .$$

On vérifie aisément que $U = u \cdot V/v$ est une solution polynomiale de cette équation.

Posons maintenant :

$$v_i = \frac{\epsilon^i v}{\mathrm{pgcd}_q(v, \epsilon v)} \qquad \text{pour } i = 0, 1,$$

on a alors :

$$z \cdot \ x^{\alpha_+} \cdot a \cdot v_0 \cdot \epsilon u - x^{\alpha_-} \cdot b \cdot v_1 \cdot u = x^{\alpha_-} \cdot b \cdot v_0 \cdot v_1 \cdot \mathrm{pgcd}_q(v, \epsilon v) \ .$$

Soit $\langle p_1, p_2, \ldots, p_m\rangle$ la qGFF de $\widehat{v}$; d'où : $v = x^\beta\, \widehat{v} = x^\beta\, [p_1]^{\frac{1}{q}}\, [p_2]^{\frac{2}{q}} \cdots [p_m]^{\frac{m}{q}}$.

On a $v_0 = \frac{v}{pgcd_q(v,\epsilon v)}$ et $v_1 = \frac{\epsilon v}{pgcd_q(v,\epsilon v)}$.

Donc $\quad v_0 = \frac{x^\beta \cdot [p_1]^{\frac{1}{q}} \cdots [p_m]^{\frac{m}{q}}}{pgcd(x^\beta,(qx)^\beta)\cdot pgcd_q(\widehat{v},\widehat{\epsilon v})} = \frac{x^\beta \cdot [p_1]^{\frac{1}{q}} \cdots [p_m]^{\frac{m}{q}}}{x^\beta \cdot [p_1]^{\frac{0}{q}} \cdots [p_m]^{\frac{m-1}{q}}} = \epsilon^0 p_1 \cdots \epsilon^{-m+1} p_m \quad ,$

et $\quad v_1 = \frac{q^\beta \cdot x^\beta \cdot [\epsilon p_1]^{\frac{1}{q}} \cdots [\epsilon p_m]^{\frac{m}{q}}}{pgcd(x^\beta,(qx)^\beta)\cdot pgcd_q(\widehat{v},\widehat{\epsilon v})} = \frac{q^\beta \cdot x^\beta \cdot [\epsilon p_1]^{\frac{1}{q}} \cdots [\epsilon p_m]^{\frac{m}{q}}}{x^\beta \cdot [p_1]^{\frac{0}{q}} \cdots [p_m]^{\frac{m-1}{q}}} = q^\beta \cdot \epsilon p_1 \cdots \epsilon p_m \quad .$

$v_0 \mid b$ et $v_1 \mid a$ donc si $P_1 := \mathrm{pgcd}_q(\epsilon^{-1}a, b)$ alors, évidemment : $p_1 \mid P_1$. Cette observation va permettre (en exploitant les propriétés de qGFF) de déterminer itérativement des multiples P_i de p_i tel que $\epsilon P_i \mid a$ et $\epsilon^{-i+1} P_i \mid b$. Finalement, on est capable de calculer :

$$\widehat{V} := [P_1]^{\frac{1}{q}}\, [P_2]^{\frac{2}{q}} \cdots [P_n]^{\frac{n}{q}} \ .$$

3.2.1 Algorithme VMULT

- ⋆ **Entrées** : a, b polynômes $\in F[x]$ respectivement premier et q-unitaire qui constituent le quotient de $\rho = z \cdot x^\alpha \cdot a/b$.
- ⋆ **Sorties** : polynômes q-unitaires $\langle P_1, P_2, \ldots, P_n\rangle$ tels que $\widehat{V} := [P_1]^{\frac{1}{q}}\,[P_2]^{\frac{2}{q}} \cdots [P_n]^{\frac{n}{q}}$ soit un multiple de $\widehat{v}$, la partie q-unitaire du dénominateur $v = x^\beta\,\widehat{v}$ de ρ.

- ETAPE 1 : Déterminer $n = \min j \in N$ tel que $\mathrm{pgcd}(\epsilon^{-1}a, \epsilon^{k-1}b) = 1$ pour tout $k > j$.
- ETAPE 2 : Poser $a_0 = a, b_0 = b$ et pour j de 1 à n :

$$P_i = \mathrm{pgcd}_q(\epsilon^{-1}a_{i-1}, \epsilon^{i-1}b_{i-1}) \ ,$$

$$a_i = a_{i-1}/\epsilon P_i \ ,$$

$$b_i = b_{i-1}/\epsilon^{-i+1}P_i \ .$$

3.2.2 Algorithme qTELESCOPING

- ⋆ **Entrée** : $\rho \in F[x]$ tel que $f_{k+1}/f_k = \rho(q^k)$ pour tout $k \in N$.
- ⋆ **Sortie** : Solution q-hypergéométrique g_k de l'équation $g_{k+1} - g_k = f_k$. [Si une telle solution n'existe pas, l'algorithme s'arrête.]

- ETAPE 1 : Décomposer ρ sous la forme $\rho = z \cdot x^\alpha \cdot a/b$ tel que $z \in F$, $x \in N$, et $a, b \in F[x]$ respectivement premier et q-unitaire.
- ETAPE 2 : Calculer $\widehat{V} := [P_1]^{\frac{1}{q}}\,[P_2]^{\frac{2}{q}} \cdots [P_n]^{\frac{n}{q}}$ par l'algorithme VMULT.
- ETAPE 3 : Déterminer :

$$\gamma := \begin{cases} log_q(z) & \text{si } \alpha = 0 \text{ et si } z \text{ est une puissance positive entière de } q \ , \\ 0 & \text{sinon.} \end{cases}$$

- ETAPE 4 : Poser $V := x^\gamma\,\widehat{V}$.
  Si l'équation $z \cdot x^{\alpha_+} \cdot a \cdot V \cdot \epsilon U - x^{\alpha_-} \cdot b \cdot U \cdot \epsilon V = x^{\alpha_-} \cdot b \cdot V \cdot \epsilon V$ admet une solution $U \in F[x]$, alors $g_k := f_k \cdot U(q^k)/V(q^k)$, sinon il n'y a pas de solution q-hypergéométrique.

3.3 q-analogue de l'algorithme de Zeilberger

Soit F une fonction q-hypergéométrique en n et k, alors il existe sous certaines conditions, une récurrence linéaire :

$$c_0(n)\,F(n,k) + c_1(n)\,F(n+1,k) + \cdots + c_d(n)\,F(n+d,k) = G(n,k+1) - G(n,k) \ ,$$

avec des coefficients polynomiaux en q^n indépendants de k et avec $G(n,k)$ une fonction rationnelle multiple de $F(n,k)$ et q-hypergéométrique en n et k.

Etant donné d l'ordre de la récurrence, $G(n,k)$ et les coefficients $c_i(n)$ sont donnés par l'algorithme *qTelescope*.

En sommant les deux membres de l'égalité sur k, on obtient :

$$c_0(n)\,a(n) + c_1(n)\,a(n+1) + \cdots + c_d(n)\,a(n+d) = 0\ ,$$

où $a(n) = \sum_k F(n,k)$.

Remarque : Axel Riese [16] a étudié les sommations hypergéométriques bibasiques, c'est-à-dire le cas où le sommant U_n vérifie $U_{n+1}/U_n = P(q^n, r^n)/Q(q^n, r^n)$.

3.4 Implémentations

Cette extension aux q-analogues a été implémentée en **MAPLE** par Doron Zeilberger, par Tom H. Koornwinder [9] et également en **MATHEMATICA** par Axel Riese [13].

1. Chargement du package **qZeil** :

```
In[1]:= <<qZeil.m
Out[1]= Axel Riese's q-Zeilberger implementation version 1.4 loaded
```

2. La formule de sommation du q-Chu-Vandermonde :

$$\sum_{k=0}^{n} \frac{(b;q)_k}{(q;q)_k}\, q^k = \frac{(bq;q)_n}{(q;q)_n}\ .$$

```
In[2]:= qZeil[qfac[b,q,k] q^k / qfac[q,q,k], {k,0,n}, n, 0]

         qfac[b q, q, n]
Out[2]= ---------------
          qfac[q, q, n]
```

3. Les polynômes de Rogers-Szegö :

$$r_n(x,a) := \sum_{k=0}^{n} \begin{bmatrix} n \\ k \end{bmatrix}_q a^{n-k}\, x^k\ ,$$

satisfont une récurrence d'ordre 2 :

```
In[3]:= qZeil[qBinomial[n,k,q] a^(n-k) x^k, {k,0,n}, n,2]

                                -1 + n
Out[3]= SUM[n] == -(a (1 - q       ) x SUM[-2 + n]) + (a + x) SUM[-1 + n]
```

4. Dennis Stanton a posé la conjecture suivante :

$$\sum_k (-1)^k\, q^{4k^2} \begin{bmatrix} 2n \\ n-4k \end{bmatrix}_q = \sum_k q^{2k^2} \begin{bmatrix} n \\ 2k \end{bmatrix}_q (-q;q^2)_{n-2k}\,(-1;q^4)_k\ . \qquad (3.3)$$

□ Le package `qZeil` permet de prouver cette conjecture sans aucune difficulté :

```
In[4]:= qZeil[ (-1)^k * q^(4*k^2) * qBinomial[2*n,n-4*k,q],
              {k,-Infinity,Infinity}, n, 3]

                    -3 + 2 n       -5 + 2 n        -4 + 2 n
Out[4]= SUM[n] == q         (1 - q        ) (1 - q        ) SUM[-3 + n]

       -7 + 2 n   5    2 n    1 + 2 n    2 + 2 n
   +  q         (q  - q    - q        - q       ) SUM[-2 + n]

        3    2 n    1 + 2 n    2 + 2 n
      (q  + q    + q        + q       ) SUM[-1 + n]
   +  --------------------------------------------
                            3
                           q

In[5]:= qZeil[q^(2*k^2) * qBinomial[n,2*k,q^2] * qfac[-q,q^2,n-2*k]
             * qfac[-1,q^4,k], {k, -Infinity,Infinity}, n, 3]

                    -3 + 2 n       -5 + 2 n        -4 + 2 n
Out[5]= SUM[n] == q         (1 - q        ) (1 - q        ) SUM[-3 + n] +

       -7 + 2 n   5    2 n    1 + 2 n    2 + 2 n
   +  q         (q  - q    - q        - q       ) SUM[-2 + n] +

        3    2 n    1 + 2 n    2 + 2 n
      (q  + q    + q        + q       ) SUM[-1 + n]
   +  --------------------------------------------
                            3
                           q
```

□ La preuve de (3.3) est complète en vérifiant les valeurs initiales des deux membres de l'égalité.

Chapitre 4

Une autre approche : Mgfun

L'objectif du projet *Algorithmes* de l'I.N.R.I.A. dirigé par Philippe Flajolet est la conception, l'évaluation et l'optimisation des algorithmes fondamentaux en informatique, dans les domaines de la transmission, du stockage et du traitement de données. Un des axes de recherche est orienté vers l'évaluation de performance des algorithmes. Celle-ci repose sur des méthodes d'analyse combinatoire et d'analyse asymptotique complexe.

C'est dans cette orientation que Frédéric Chyzak a conçu le package `Mgfun`. C'est un outil de calcul formel sur les fonctions usuelles et spéciales.

Précisemment, il permet :

- les opérations d'anneau (addition, produit, pgcd, ...) dans des algèbres non commutatives;
- l'élimination dans ces algèbres (notamment par calcul des bases de Gröbner);
- les opérations sur les fonctions holonomes (produit, sommation).

Ce package est utilisé pour effectuer des calculs sur les fonctions à plusieurs variables. En particulier, pour les fonctions spéciales et les fonctions hypergéométriques, leurs q-analogues et les fonctions holonomes en général.

Il est ainsi possible de déterminer des formes closes (en terme de fonctions hypergéométriques et q-hypergéométriques) et également de vérifier des identités.

4.1 Systèmes holonomes

4.1.1 Classe des fonctions holonomes (D-finies)

Différentes définitions :

1. Une fonction f à plusieurs variables $x_1, x_2, \ldots, x_d$ est D-finie s'il existe d polynômes P_i à coefficients rationnels tels que pour tout $i = 1, 2, \ldots, d$:

$$P_i(D_i)\, f(x_1, x_2, \ldots, x_d) = 0 \ ,$$

   où D_i est l'opérateur de dérivation par rapport à la variable x_i.

2. Une fonction f est D-finie si ses dérivées successives engendrent un espace vectoriel de dimension finie.

4.1.2 Classe des suites holonomes (P-récursives)

Différentes définitions :

1. Une suite a_n est P-récursive s'il existe un polynôme P à coefficients rationnels tel que $P(E)\, a_n = 0$ où E est l'opérateur de décalage suivant : $E\, a_n = a_{n+1}$.

2. Une suite a_n est P-récursive si ses décalées successives engendrent un espace vectoriel de dimension finie.

4.1.3 Systèmes holonomes

Une fonction D-finie est totalement déterminée par des équations différentielles et par ses conditions initiales, tandis qu'une suite P-récursive est déterminée par des équations de récurrence et par ses premiers termes. Dans les deux cas, la quantité d'information nécessaire à la détermination est finie.

La définition des deux concepts précédemment introduits implique que les algorithmes utilisés pour d'une part les fonctions D-finies, et d'autre part les suites P-récursives sont similaires.

On définit alors une *fonction holonome* par un système fini d'équations mixtes (équations différentielles ou récurrences linéaires), et par un nombre fini de conditions initiales.

4.2 Algèbre de Weyl, Algèbre de Ore

4.2.1 Algèbre de Weyl

Etant donnés deux couples d'indéterminées $(x_1, \ldots, x_d)$ et $\partial = (\partial_1, \ldots, \partial_d)$ appartenant à un ensemble $\mathbb{K}$, une algèbre de *Weyl* est définie comme l'anneau non commutatif des polynômes $\mathbb{K}\langle x, \partial\rangle = \mathbb{K}\langle x_1, \ldots, x_d, \partial_1, \ldots, \partial_d\rangle$ avec les règles de commutation suivantes :

$$\partial_i x_j = x_j \partial_i + \delta_{i,j} \ ,$$

$$\partial_i \partial_j = \partial_j \partial_i \ ,$$

$$x_i x_j = x_j x_i \ ,$$

où $\delta_{i,j}$ est le symbole de Kronecker.

4.2.2 Algèbre de Ore et systèmes holonomes

Etant donnés deux couples d'indéterminées $(x_1, \ldots, x_d)$ et $\partial = (\partial_1, \ldots, \partial_d)$ appartenant à un ensemble $\mathbb{K}$, une algèbre de *Ore* est définie comme l'anneau non commutatif des polynômes $\mathbb{K}\langle x, \partial\rangle = \mathbb{K}\langle x_1, \ldots, x_d, \partial_1, \ldots, \partial_d\rangle$ avec les règles de commutation suivantes :

$$\partial_i x_j = \sigma_i(x_j)\partial_i + \delta_i(x_j) \ ,$$

$$\partial_i \partial_j = \partial_j \partial_i \ ,$$

$$x_i x_j = x_j x_i \ ,$$

où les $\sigma(i)$ et les $\delta(i)$ sont respectivement des morphismes d'anneaux et des morphismes linéaires de $\mathbb{K}[x_i]$ dans $\mathbb{K}[x]$ et tels que $\delta_i(fg) = \sigma_i(f)\, \delta_i(g) + \delta_i(f)\, g$.

Ces définitions permettent de montrer que l'algèbre de *Ore* est un concept qui généralise les algèbres de *Weyl*.

En effet, posons $\sigma(x) = x$ et $\delta(x) = 1$, alors $\partial x = x\partial + 1$. Dans ce cas, l'algèbre de *Ore* est l'algèbre de *Weyl* à une seule variable. L'opérateur ∂ peut être vu comme la dérivée D_x sur $\mathbb{K}[x]$ et $\mathbb{K}\langle x, \partial\rangle = \mathbb{K}\langle x, D_x\rangle$.

D'autre part, on peut maintenant utiliser ces algèbres pour étudier les systèmes holonomes. En voici un exemple :

Posons $\sigma(x) = x + 1$ et $\delta(x) = 0$, alors $\partial x = (x + 1)\,\partial$ et ∂ est l'opérateur de décalage. Changement de notation ($n = x$ et $S_n = \partial$) : $S_n\, n = (n + 1)\, S_n$ et $\mathbb{K}\langle x, \partial\rangle = \mathbb{K}\langle n, S_n\rangle$.

Voici quelques exemples de fonctions holonomes (avec une description en terme d'opérateurs) :

- Factorielle : Soit $u_n = n!$ alors $(S_n - (n+1))\, u = 0$. Ceci montre que $n!$ est P-récursive.
- Soit f, la suite des fonctions $F_n(x) = n!\, \cos x$. Dans l'algèbre de *Ore* $\mathbb{K}\langle x, n, D_x, S_n\rangle$, on a $(D_x^2 + 1)\, f = 0$ et $(S_n - (n + 1))\, f = 0$.

4.3 Application

Ces définitions sont les bases théoriques de l'implémentation de `Mgfun`. En effet, sa particularité est de pouvoir construire des algèbres en définissant les règles de commutation. Voici un exemple concret d'utilisation du package :

- ☐ Chargement du package :

```
> with(Mgfun):
```

- ☐ Le sommant considéré est le coefficient du binôme :

```
> h:=binomial(n,k):
```

- ☐ Voici l'algèbre dans laquelle on effectue les calculs :

```
> A:=shiftalg([Sn,n],[Sk,k]):
```

  Voici les équations satisfaites par les coefficients binomiaux (système holonome) :

```
> G:=hypergeomtoholon(h,A);
                  [(n + 1 - k) Sn - n - 1, (-k - 1) Sk + n - k]
```

- ☐ Élimination :

```
> skewelim(op(G),k,A);
                                Sn Sk - 1 - Sk
```

  Creative telescoping :

```
> subs(Sk=1,");
                                   Sn - 2
```

□ Nous appliquons maintenant l'élimination par bases de Gröbner qui est une méthode plus générale. Dans un premier temps, il faut tester le caractère holonome de la fonction (vérifier que les équations dans G fournissent suffisamment d'informations pour éliminer une variable). Pour cela, on introduit un ordre de termes sur toutes les lettres et on calcule la dimension de Hilbert :

```
> T:=termorder(A,tdeg=[Sn,Sk,n,k]):

> hilbertdim(G,T);
                                       2
```

On trouve 2, autant que d'opérateurs (ici, S_n et S_k), donc l'étape suivante va retourner un résultat intéressant (c'est-à-dire non nul). On introduit donc un nouvel ordre de termes pour éliminer k et ne conserver que S_n et S_k.

```
> T:=termorder(A,lexdeg=[[k],[Sn,Sk]]):
```

Élimination :

```
> GB:=gbasis(G,T):
        [-Sk k - Sk + n - k, Sn n + Sn - Sn k - n - 1, Sn Sk - 1 - Sk]

> nops(subs(Sk=1,remove(has,GB,k)));
                                       1
```

Et "creative telescoping" :

```
> collect(primpart(op(subs(Sk=1,remove(has,GB,k))),Sn),Sn,factor);
                                     Sn - 2
```

Annexe A

Les Packages

Les différents packages qui traitent des problèmes énumérés dans ce rapport, sont présentés et les fonctions principales sont décrites précisément.

A.1 Package `EKHAD`

Doron Zeilberger, Mai 1996, *Maple*
Démonstration d'identités hypergéométriques.

□ `zeil(SUMMAND,k,n,N)`
`zeil(SUMMAND,k,n,N,MAXORDER)`
`zeil(SUMMAND,k,n,N,MAXORDER,parameter_list)`

→ **retourne une équation de récurrence linéaire à coefficients polynomiaux;**

- Le sommant doit être un produit de factorielles et/ou de coefficients binomiaux et/ou de factorielles montantes $(a)_k$ [notés `rf(a,k)`] et/ou des puissances en k et en n, et d'un facteur polynomial optionnel;
- k est la variable de sommation, n est la variable auxiliaire;
- L'opérateur de décalage en n est noté `N`;
- L'ordre maximal de la récurrence retournée est `MAXORDER` (par défaut : 6);

La fonction `zeil` retourne $ope(N,n)$ et $R(n,k)$ tel que

$$ope(N,n)\ SUMMAND = G(n,k+1) - G(n,k)\ ,$$

où $G(n,k) = R(n,k)\ SUMMAND$.

□ `zeilpap(SUMMAND,k,n)`

Idem `zeil` avec une "preuve rédigée";

□ `zeillim(SUMMAND,k,n,N,alpha,beta)`

Idem `zeil`, mais détermine une récurrence pour la somme de $k = alpha$ à $k = n + beta$.

A.2 Package qEKHAD

Doron Zeilberger, Avril 1996, *Maple*
Démonstration d'identités q-hypergéométriques.

□ `qzeil(SUMMAND,N,k,n,para)`

→ **retourne une équation de récurrence linéaire à coefficients polynomiaux;**

- Le sommant doit être un produit de q-factorielles de la forme :

```
qfac(r):=(1-q)*(1-q^2)...(1-q^r)
qbin(a,b):=qfac(a)/(qfac(b)*qfac(b-a))
qrf(a,k):=(1-a)*(1-q*a)...(1-q^(k-1)*a)
```

- L'opérateur de décalage en n est noté `N` ;
- k est la variable de sommation, n est la variable auxiliaire;
- `para` est la liste des autres variables discrètes;
- L'ordre maximal de la récurrence retournée est `MAXORDER` (par défaut : 6);

La fonction `qzeil` retourne $ope(N,n)$ et $R(n,k)$ tel que

$$ope(N,n)\ SUMMAND = G(n,k+1) - G(n,k)\ ,$$

où $G(n,k) = R(n,k)\ SUMMAND$.

□ `qzeilpap(SUMMAND,k,n,para)`

Idem `qzeil` avec une “preuve rédigée”;

A.3 Package zeilb

Tom H. Koornwinder, Septembre 1993, *Maple*
Sommation de fonctions hypergéométriques.

```
zeilb([a1, a2, ... ], [b1, b2, ... ], z, n, l, t)
```

- `a1, a2, ...` et `b1, b2, ...` sont respectivement les paramètres du numérateur et du dénominateur de la série;
- `l` est l'ordre supposé de la récurrence;
- `t` est une variable optionnelle qui permet d'afficher les calculs intermédiaires lors de l'exécution;
- La fonction `zeilb` tente d'évaluer (si $l = 0,1$) la série

$$\sum_{k=0}^{n} \frac{(a_1)_k\ (a_2)_k \dots}{(b_1)_k\ (b_2)_k \dots} \frac{z^k}{k!}$$

  sinon, retourne une récurrence exprimant `INH(n)` en fonction de `INH(n-1), ...` où `INH(n)` est la série donnée en argument.

A.4 Package qzeilb

Tom H. Koornwinder, Septembre 1993 ***Maple***
Sommation de fonctions q-hypergéométriques.

□ `qzeilb([a1, a2, ... ], [b1, b2, ... ], q, z, n, l, t)`

- `a1, a2, ...` et `b1, b2, ...` sont respectivement les paramètres du numérateur et du dénominateur de la série;
- `l` est l'ordre supposé de la récurrence;
- `t` est une variable optionnelle qui permet d'afficher les calculs intermédiaires lors de l'éxecution;
- La fonction `qzeilb` tente d'évaluer (si $l = 0, 1$) la série

$$\sum_{k=0}^{n} \frac{(a_1;q)_k\,(a_2;q)_k \ldots}{(b_1;q)_k\,(b_2;q)_k \ldots (q;q)_k} \left((-1)^k\, q^{k(k-1)/2}\right)^{r-s+1} z^k$$

  sinon, retourne une récurrence exprimant `INQH(n)` en fonction de `INQH(n-1)`, `INQH(n-2), ...` où `INQH(n)` est la série donnée en argument.

A.5 Package ZEILBERG

Wolfram Koepf, ***Reduce***

□ `gosper(a,k)`
`gosper(a,k,m,n)`

- `gosper(a,k)` détermine (si elle existe) une forme T telle que $T_k - T_{k-1} = a_k$ (algorithme de Gosper);
- `gosper(a,k,m,n)` détermine $\sum_{k=m}^{n} a_k$ en appliquant l'algorithme de Gosper;

□ `sumrecursion(f,k,n)`
`sumrecursion(f,k,n,j)`

- Cette fonction est l'implémentation de l'algorithme rapide de Zeilberger. Elle retourne une équation de récurrence pour :

$$sum(n) = \sum_{k=-\infty}^{\infty} f(n,k)$$

- `j` est l'ordre de la récurrence cherchée (paramètre optionnel);

□ `hyperrecursion(upper,lower,x,n)`
`hyperrecursion(upper,lower,x,n,j)`

- Cette fonction détermine une équation de récurrence par rapport à n pour la série

$${}_pF_q \left[\begin{matrix} a_1, \ldots, a_p \\ b_1, \ldots, b_q \end{matrix}; x\right] ,$$

  où `upper`$= \{a_1, a_2, \ldots, a_p\}$ et `lower`$= \{b_1, b_2, \ldots, b_q\}$;
- `j` est l'ordre de la récurrence cherchée (paramètre optionnel);

A.6 Package `Zeil`

Peter Paule et Markus Schorn, Juin 1993, *Mathematica*

□ `Gosper[SUMMAND,RANGE,ORDER]`

- Cette fonction est une extension de l'algorithme de Gosper qui détermine la forme close d'une somme.

□ `Zb[SUMMAND,RANGE,RECVAR,ORDER]`

- Cette fonction est l'implémentation de l'algorithme rapide de Zeilberger qui cherche une récurrence satisfaite par la série.

A.7 Package `qZeil`

Peter Paule et Axel Riese, *Mathematica*

□ `qTelescope[SUMMAND,RANGE]`

- Cette fonction est une extension de l'algorithme de Gosper aux cas des fonctions q-hypergéométriques.

□ `qZeil[SUMMAND,RANGE,RECVAR,ORDER]`

- Cette fonction est l'implémentation de l'extension de l'algorithme rapide de Zeilberger pour les series q-hypergéométriques qui cherche une récurrence satisfaite par la série.

□ Ce package permet également la recherche des identités de type "companion" et des paires q-WZ duales.

A.8 Package `HYP` & `HYPQ`

Christian Krattenthaler, Juin 1995, *Mathematica*

Ces packages (c.f. [10]) ne traitent pas directement des problèmes de sommation de séries hypergéométriques. Néanmoins, ce sont des outils très utiles pour manipuler facilement les expressions faisant intervenir des termes hypergéométriques (et q-hypergéométriques) et des coefficients binomiaux (et q-binomiaux). L'idée est de laisser le contrôle de chaque étape des transformations à l'utilisateur en le guidant dans ces manipulations. (Le package ne permet pas de transformer automatiquement une série).

□ Chargement du package

```
In[1]:= <<hyp.m
```

□ On considère la somme de Vandermonde :

```
In[2]:= Sum[Binomial[N,l]*Binomial[M,K-l],{l,0,Infinity}]

         Inf
        -----(       ) (   )
         \   (   M   ) ( N )
Out[2]=   >  (       ) (   )
         /   ( K - l ) ( l )
        -----(       ) (   )
         l=0
```

que l'on convertit en notation hypergéométrique :

```
In[3]:= %/.SumF

           [              ]
           |  -K, -N      |
         F |          ; 1 | (1 - K + M)
        2 1| 1 - K + M    |            K
           [              ]
Out[3]= --------------------------------
                     (1)
                        K
```

☐ Le package HYP inclut de nombreuses formules de sommation. L'étape suivante nous permet de connaitre celles qui sont applicables à notre somme.

```
In[4]:= %/.SListe
Is N a nonnegative integer?
[y|n]:  y

Be sure to apply "FOrdne" before using the following information!

           S2101 (1 - K + M)      S2103 (1 - K + M)
                            K                      K
Out[4]= {{------------------}, {------------------}}
                 (1)                   (1)
                    K                     K
```

les formules directement applicables sont les suivantes :

```
In[5]:= Sgl2101
Do you want to set values for the equation? [y|n]: n

           [          ]    (-a + c)
           | a, -n    |            n
Out[5]=  F |      ; 1 | == ---------
        2 1|   c      |      (c)
           [          ]         n

In[6]:= Sgl2103
Do you want to set values for the equation? [y|n]: n

           [         ]      [                ]
           | a, b    |      | c, -a - b + c  |
Out[6]=  F |     ; 1 | == Ga|                |
        2 1|  c      |      | -a + c, -b + c |
           [         ]      [                ]
```

□ On applique alors ces formules, ce qui nous donne deux réécritures de notre formule initiale :

```
In[7]:= %3/.FOrdne/.S2101
Is N a nonnegative integer?
[y|n]:  y

        (1 + M)  (1 - K + M)
               N            K
Out[7]= ---------------------
          (1)  (1 - K + M)
             K            N

In[8]:= %3/.FOrdne/.S2103

          [                      ]
          | 1 - K + M, 1 + M + N |
        Ga|                      | (1 - K + M)
          | 1 + M, 1 - K + M + N |            K
          [                      ]
Out[8]= ---------------------------------------
                         (1)
                            K
```

□ Un autre exemple : considérons la série hypergéométrique suivante :

```
In[9]:= F[{a, b, c + n}, {c, 1 + a}, 1]

           [                 ]
           | a, b, c + n     |
Out[9]=  F |             ; 1 |
        3 2|  c, 1 + a       |
           [                 ]
```

□ On cherche les formule de sommation qui peuvent s'appliquer à notre expression :

```
In[10]:= %/.SListe

Out[10]= {}
```

□ On obtient un ensemble vide, on cherche alors les formules de transformation :

```
In[11]:= %9/.TListe
Is -c - n a nonnegative integer?
[y|n]:  n
Is -b a nonnegative integer?
[y|n]:  n
Is -a a nonnegative integer?
[y|n]:  n
Is -c - n a nonnegative integer?
[y|n]:  n
Is -b a nonnegative integer?
[y|n]:  n
Is -a a nonnegative integer?
[y|n]:  n
Is -c - n a nonnegative integer?
[y|n]:  n
Is -b a nonnegative integer?
[y|n]:  n
Is -a a nonnegative integer?
[y|n]:  n

Be sure to apply "FOrdne" before using the following information!

Out[11]= {{T3204}, {T3205}, {T3261}, {T3262}, {T3263}, {T3264}, {T3267}, {T3268}}
```

□ On applique la première qui est proposée :

```
In[12]:= Tgl3204
Do you want to set values for the equation? [y|n]: n

            [            ]       [                      ]
            | a, b, c    |       | a, -b + d, -c + d    |
Out[12]=  F |        ; 1 | ==  F |                 ; 1 |
         3 2|  d, e      |    3 2| d, -b - c + d + e    |
            [            ]       [                      ]

        [                      ]
        | e, -a - b - c + d + e  |
>     Ga|                      |
        | -a + e, -b - c + d + e |
        [                      ]

In[13]:= %9/.FOrdne/.T3204

            [                     ]   [                  ]
            |  a, -b + c, -n      |   | 1 + a, 1 - b - n |
Out[13]=  F |                ; 1  | Ga|                  |
         3 2| c, 1 + a - b - n    |   | 1, 1 + a - b - n |
            [                     ]   [                  ]
```

□ Maintenant, quelles formules de sommation peut-on appliquer ?

```
In[14]:= %/.SListe
Is n a nonnegative integer?
```

```
[y|n]:  y

Be sure to apply "FOrdne" before using the following information!

                    [                   ]           [                   ]
                    | 1 + a, 1 - b - n  |           | 1 + a, 1 - b - n  |
Out[14]= {{S3201 Ga|                   |}, {S3261 Ga|                   ]}}
                    | 1, 1 + a - b - n  |           | 1, 1 + a - b - n  |
                    [                   ]           [                   ]
```

☐ On applique la première :

```
In[15]:= %13/.FOrdne/.S3201
Is n a nonnegative integer?
[y|n]:  y

            [                  ]
            | 1 + a, 1 - b - n |
          Ga|                  | (b)  (-a + c)
            | 1, 1 + a - b - n |    n         n
            [                  ]
Out[15]= -------------------------------------
                    (-a + b)  (c)
                            n    n
```

A.9 Package `Mgfun 1.2`

Frédéric Chyzak, 1996, *Maple*

Ce package est un ensemble de fonctions effectuant des calculs sur les fonctions holonomes, D-finies, P-finies. Le nom MGFUN [Multivariate Generating FUNctions] est essentiellement historique puisqu'en fait, ce package permet :

- la manipulation d'opérateurs linéaires;
- le calcul de bases de Gröbner non commutatives;
- la manipulation des fonctions holonomes, ainsi que des classes similaires, dont des extensions au q-calcul.

(c.f. chapitre 4 page 33).

Conclusion

Une *identité hypergéométrique* est typiquement la somme d'une énorme expression faisant intervenir des coefficients binomiaux, des factorielles, des fonctions rationnelles et des fonctions puissances, et qui se simplifie miraculeusement.

Nous avons présenté dans ce rapport les principales méthodes élaborées dans le domaine de la manipulation de ces séries. D. Zeilberger a popularisé et implémenté des algorithmes de démonstration de relations de récurrence satisfaites par des fonctions hypergéométriques (algorithmes de Gosper, de Zeilberger, méthode du "creative telescoping").

Ces algorithmes ont permis la démonstration automatique d'identités non triviales (identités de Rogers-Ramanujan, formule du q-Chu-Vandermonde, ...) ainsi que la découverte de nouvelles identités grâce à de nouvelles techniques ("companion identity", identité duale, ...). De façon générale, ces méthodes permettent de démontrer toutes identités hypergéométriques.

Le développement des systèmes de calcul formel permet la manipulation et l'expérimentation des objets sortant du cadre traditionnel de la résolution symbolique d'équations algébriques ou différentielles. On peut dorénavant calculer sur des structures de plus en plus élaborées (extensions algébriques, algèbres non commutatives ou à commutativité tordue). Ainsi, de nombreux outils pour manipuler les séries hypergéométriques ont déjà été élaborés et implémentés notamment par C. Krattenthaler, D. Zeilberger, P. Paule et F. Chyzak dans les environnements MATHEMATICA et MAPLE.

La démonstration d'une identité est généralement un enchaînement de transformations et de spécialisations des arguments. Cet aspect peut être automatisé soit comme outil de démonstration automatique (c.f. [15]) ou, plus modestement, comme aide au calcul (c.f. [10]). Ceci est encore plus vrai pour le cas des fonctions hypergéométriques basiques, où la présence de paramètres supplémentaires rend les calculs plus complexes.

Ainsi, il est intéressant d'envisager des bibliothèques permettant une aide générale et efficace à l'utilisateur pour les séries hypergéométriques mais également pour les q-séries. Ceci nécessite, en particulier, de s'intéresser aux propriétés de ces séries et d'explorer en détail les différentes techniques déjà développées par les spécialistes du sujet.

Les séries hypergéométriques bibasiques sont une autre extension intéressante, celles-ci apparaissent dans [6, 16] et nécessitent de lourds calculs.

Il existe aussi un lien entre fonctions hypergéométriques et fonctions symétriques exploité par J. Désarménien [3]. Des algorithmes de manipulation de celles-ci ont été mis sous forme d'une série de bibliothèques MAPLE par S. Veigneau [17]. Cela permet d'espérer l'enrichissement des outils de manipulation des fonctions hypergéométriques.

Bibliographie

[1] P. Cartier, *Démonstration "automatique" d'identités et fonctions hypergéométriques [d'après D. Zeilberger]*, Séminaire Bourbaki, **746** (1991-92).

[2] F. Chyzak, *Holonomic systems and automatic proofs of identities*, Institut National de Recherche en Informatique et en Automatique, Research Report **2371** (1994).

[3] J. Désarménien, *Fonctions symétriques et séries hypergéométriques basiques multivariées*, Séminaire Lotharingien de Combinatoire, **B11a** (1984).

[4] S. B. Ekhad et J. E. Majewicz, *A short WZ-style proof of Abel's identity*, Electronic journal of combinatorics, volume 3 (2), **R16** (1996).

[5] S. B. Ekhad et S. Tre, *A purely verification of the first Rogers-Ramanujan identity*, Journal of combinatorial theory (Series A), **54** (1990), 309–311.

[6] G. Gasper et M. Rahman, *Basic hypergeometric series*, Encyclopedia of Mathematics and its applications, Cambridge University Press, Cambridge, **35** (1990).

[7] R. W. Gosper, Jr, *Decision procedure for indefinite hypergeometric summation*, Proceedings of the national academy of sciences of the United States of America, **75** (1978), 40–42.

[8] W. Koepf, *A REDUCE package for indefinite and definite summation*, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Technical Report **94-33** (1994).

[9] Tom H. Koornwinder, *On Zeilberger's algorithm and its q-analogue*, Journal of computational and applied mathematics, **48** (1993), 91–111.

[10] C. Krattenthaler, *`HYP` and `HYPQ`, Mathematica packages for the manipulation of binomial sums and hypergeometric series, respectively q-binomial sums and basic hypergeometric series*, Journal of symbolic computation, **20** (1995), 737–744. (Les packages sont disponibles sur la page WWW `http://radon.mat.univie.ac.at/People/kratt/`).

[11] P. Paule, *Greatest Factorial Factorization and Symbolic Summation*, Journal of symbolic computation, **20** (1995), 235–268.

[12] P. Paule et M. Schorn, *A Mathematica version of Zeilberger's algorithm for proving binomial coefficient identities*, RISC-Linz Report Series, **93-36** J. Kepler University, Linz (1993).

[13] P. Paule et A. Riese, *A Mathematica q-analogue of Zeilberger's algorithm based on an algebraically motivated approach to q-hypergeometric telescoping*, (1991).

[14] M. Petkovsek et B. Salvy, *Finding all hypergeometric solutions of linear differential equations*, Institut National de Recherche en Informatique et en Automatique, Research Report **1907** (1993).

[15] M. Petkovsek, H. S. Wilf et D. Zeilberger, $A = B$, A.K. Peters, Wellesley, Massachusetts, (1996). (Les programmes sont disponibles sur le site WWW `http://www.cis.upenn.edu/~wilf/AeqB.html`).

[16] A. Riese, *A generalization of Gosper's algorithm to bibasic hypergeometric summation*, Electronic journal of combinatorics, volume 3, **R19** (1996).

[17] S. Veigneau, *ACE, an Algebraic Combinatorics Environment for the computer algebra system MAPLE, User's Reference Manual, Version 1.0*, Rapport de l'Institut Gaspard Monge, **95-14** (1995).

[18] H.S. Wilf et D. Zeilberger, *Rational functions certify combinatorial identities*, Journal of the american mathematical society, **3** (1990), 147–158.

[19] H.S. Wilf et D. Zeilberger, *An algorithmic proof theory for hypergeometric (ordinary and "q") multisum/integral identities*, Inventiones mathematicæ, **108** (1992), 575–633.

[20] H.S. Wilf et D. Zeilberger, *Rational functions certification of hypergeometric multisum/integral/"q" identities*, Bulletin of the american mathematical society (new series), **27** (1992), 148–153.

[21] D. Zeilberger, *A holonomic systems approach to special function identities*, Journal of computational and applied mathematics, **32** (1990), 321–368.

[22] D. Zeilberger, *Three recitations on holonomic systems and hypergeometric series*, Séminaire Lotharingien de Combinatoire, **B24a** (1990).

[23] D. Zeilberger, *The method of creative telescoping*, Journal of symbolic computation, **11** (1991), 195–204.

[24] D. Zeilberger, *Theorems for a price : tomorrow's semi-rigorous mathematical culture*, Notices of the american mathematical society, **40** (1993), 978–981.