Introduction au Calcul formel

Algorithmes et complexité

Sommation symbolique :
Algorithmes sur les récurrences linéaires

Frédéric Chyzak

frederic.chyzak@inria.fr

http://www.enseignement.polytechnique.fr/profs/informatique/Frederic.Chyzak/

Avertissement

Cette séance traite la sommation symbolique du point de vue « solutions de récurrences linéaires à coefficients polynomiaux ».

Tout ce qui suit se transpose à l'intégration symbolique.

À l'inverse, le point de vue « liouvillien » de l'algorithme de Risch se transpose à la sommation symbolique (algorithme de Karr).

Suites et sommes : rappel de terminologie

Une suite $(u_n)_{n\in\mathbb{N}}$ associe une valeur à chaque entier n.

La série de terme général u_n, notée $\sum_{k\geq 0} u_k$, est la suite $(U_n)_{n\in\mathbb{N}}$ des sommes partielles

$$U_n = u_0 + \cdots + u_n = \sum_{k=0}^{n} u_k.$$

La série converge lorsque U_n a une limite S, notée alors $\sum_{k=0}^{\infty} u_k$.

U_n est une somme indéfinie de u_n ; elle vérifie $U_n - U_{n-1} = u_n$.

La somme S de la série est la somme définie des u_n sur $\mathbb{N}$.

Sommation symbolique : sommes indéfinies

Si u_n est donné sous forme close (= dans une classe bien définie) :

- polynomiale, $u_n = P(n)$ pour $P \in \mathbb{C}[X]$,
- rationnelle, $u_n = R(n)$ pour $R \in \mathbb{C}(X)$,
- hypergéométrique, $u_{n+1}/u_n = R(n)$ pour $R \in \mathbb{C}(X)$,
- polynomialement récursive,

$$P_d(n)u_{n+d} + \cdots + P_0(n)u_n = 0 \qquad \text{pour des } P_i \in \mathbb{C}[X],$$

- par des sommes emboîtées selon certaines contraintes (alembertienne, liouvillienne, etc),

déterminer s'il existe une forme close U_n telle que $U_n - U_{n-1} = u_n$.

Exemple : sommation indéfinie hypergéométrique

$$\sum_{k=0}^{n} \boxed{\frac{4^k}{\binom{2k}{k}}} = \frac{2(n+1)}{3} \boxed{\frac{4^n}{\binom{2n}{n}}} + \frac{1}{3}.$$

$$b_{n,m} = \binom{n}{m} = \text{coefficient binomial} = \frac{n!}{m!\,(n-m)!}$$

$$\Rightarrow \quad \frac{b_{n+1,m}}{b_{n,m}} = \frac{n+1}{n+1-m}, \quad \frac{b_{n,m+1}}{b_{n,m}} = \frac{n-m}{m+1}.$$

$$u_n = \frac{4^n}{\binom{2n}{n}} \quad \Rightarrow \quad \frac{u_{n+1}}{u_n} = \frac{2(n+1)}{2n+1}.$$

Exemple : sommation indéfinie polynomialement récursive

$$\sum_{k=p+1}^{n} \boxed{\binom{k}{p} H_k} =$$

$$\frac{(n+1)^2}{(p+1)^2} \boxed{\binom{n}{p} H_n} - \frac{(n-p)(n-p+1)}{(p+1)^2} \boxed{\binom{n+1}{p} H_{n+1}}$$

$$= \binom{n}{p} \left(\frac{n+1}{p+1} H_n - \frac{n-p}{(p+1)^2} \right).$$

$$H_n = \text{nombre harmonique} = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}, \qquad u_n = \binom{n}{p} H_n$$

$$\Rightarrow \quad (n+1-p)(n+2-p)u_{n+2} - (2n+3)(n+1-p)u_{n+1} + (n+1)^2 u_n = 0.$$

Sommation symbolique : sommes définies

Constantes : difficile, peu (pas ?) d'algorithmes de sommation.

$$\sum_{n=1}^{\infty} \frac{1}{n^s} = \zeta(s), \qquad \zeta(2p) \in \mathbb{Q}\, \pi^{2p}.$$

Sommes paramétrées : si u est donné sous forme close (hypergéométrique, ∂-finie), déterminer respectivement

$$U_n = \sum_{k=0}^{\infty} u_{n,k} \qquad \text{ou} \qquad U(z) = \sum_{k=0}^{\infty} u_k(z)$$

sous forme close.

Exemples : sommation définie hypergéométrique

$$\sum_{k=0}^{\infty}(-1)^k\binom{2n}{k}\binom{2k}{k}\binom{4n-2k}{2n-k}=\sum_{k=0}^{2n}\cdots=\binom{2n}{n}^2.$$

Exemple : sommation définie ∂-finie (1)

$$\sum_{k=0}^{\infty}\left(\sum_{j=0}^{k}\binom{n}{j}\right)^3 = n2^{3n-1} + 2^{3n} - 3n2^{n-2}\binom{2n}{n} \qquad \text{(Calkin, 1994).}$$

$$v_{n,k} = \sum_{j=0}^{k}\binom{n}{j} \quad \Rightarrow \quad \text{les } v_{n+i,k+j} \text{ se récrivent sur } (v_{n,k}, v_{n,k+1}) :$$

$$(k+2)v_{n,k+2} - (n+1)v_{n,k+1} + (n-k-1)v_{n,k} = 0,$$

$$(k+1)v_{n,k+1} + (n-k)v_{n+1,k} - (2n+1-k)v_{n,k} = 0.$$

$$u_{n,k} = v_{n,k}^3 \quad \Rightarrow$$

$$\text{les } u_{n+i,k+j} \text{ se récrivent sur } (u_{n,k}, u_{n+1,k}, u_{n,k+1}, u_{n,k+2}).$$

Exemple : sommation définie ∂-finie (2)

$$\sum_{k=-\infty}^{+\infty} J_k(z)^2 = 1 \qquad \text{(Neumann)}.$$

$$J_k(z) = \text{fonctions de Bessel} = \left(\frac{z}{2}\right)^k \sum_{n=0}^{\infty} \frac{(-z^2/4)^n}{n!\,(n+k)!} \qquad \Rightarrow$$

les $J_{k+i}^{(j)}(z)$ se récrivent sur $\big(J_k(z), J_k'(z)\big)$ ou sur $\big(J_k(z), J_{k+1}(z)\big)$:

$$z^2 J_k''(z) + zJ_k'(z) + (z^2 - k^2)J_k(z) = 0, \qquad zJ_k'(z) + zJ_{k+1}(z) - kJ_k(z) = 0,$$

$$zJ_{k+2}(z) - 2(k+1)J_{k+1}(z) + zJ_k(z) = 0.$$

$$u_k(z) = J_k(z)^2 \qquad \Rightarrow$$

les $u_{k+1}^{(j)}(z)$ se récrivent sur $\big(u_k(z), u_{k+1}(z), u_k'(z)\big)$.

Méthodes et algorithmes principaux

1. Suites hypergéométriques :
   - Méthode de Fasenmyer pour les sommes définies
   - Algorithme de Gosper pour les sommes indéfinies
   - Algorithme de Zeilberger pour les sommes définies
2. Suites ∂-finies :
   - Bases de Gröbner pour les sommes indéfinies et définies
   - Algorithmes de Chyzak pour les sommes indéfinies et définies
3. Sommes dans des tours de corps à différences :
   - Algorithme de Karr pour les sommes indéfinies

Creative Telescoping

Soit à évaluer la somme

$$U_n = \sum_{k=a}^{b} u_{n,k} \qquad a \in \mathbb{Z} \cup \{-\infty\}, \quad b \in \mathbb{Z} \cup \{+\infty\}.$$

Supposons que l'on détermine une relation de la forme

$$\eta_d(n) u_{n+d,k} + \cdots + \eta_0(n) u_{n,k} = v_{n,k+1} - v_{n,k}.$$

Alors, après sommation sur k entre a et b, on trouve

$$\eta_d(n) U_{n+d} + \cdots + \eta_0(n) U_n = v_{n,b+1} - v_{n,a}.$$

Sommation à bornes naturelles : quand $v_{n,a} = v_{n,b+1}$.

Suites hypergéométriques

Une suite $f = (f_{n,k})$ est dite hypergéométrique si

$$\frac{f_{n-1,k}}{f_{n,k}} \in \mathbb{C}(n,k) \qquad \text{et} \qquad \frac{f_{n,k-1}}{f_{n,k}} \in \mathbb{C}(n,k).$$

Exemples : produits et quotients de factorielle et de coefficients binomiaux et multinomiaux ; factorielles montantes et descendantes.

La méthode de Fasenmyer

ENTRÉE : terme hypergéométrique $f_{n,k}$.
SORTIE : récurrence linéaire sur f à coefficients polynomiaux ne faisant pas intervenir k.

MÉTHODE :

1. Pour un ensemble de décalages S donné, on cherche à résoudre
$$\sum_{(i,j)\in S} c_{i,j}(n) f_{n-i,k-j} = 0 \ ;$$

2. Après division par $f_{n,k}$ et après avoir chassé les dénominateurs, on trouve des polynômes $p_{i,j}$ tels que
$$\sum_{(i,j)\in S} c_{i,j}(n) p_{i,j}(n,k) = 0 \ ;$$

3. Égaler les coefficients des k^l à 0 et résoudre le système linéaire en les $c_{i,j}$ ainsi obtenus ;

4. On obtient

$$\sum_{(i,j)\in S} c_{i,j}(n) f_{n-i,k} = g_{n,k} - g_{n,k-1}$$

   pour un g adéquat ;

5. Après sommation en k et introduction de $F_n = \sum_{k=a}^{b} f_{n,k}$,

$$\sum_{(i,j)\in S} c_{i,j}(n) F_{n-i} = g_{n,b} - g_{n,a-1}.$$

Améliorations : optimisation de S ; multisommes et dépendance polynomiale des $c_{i,j}$ en k.

Identité de Dixon

Sous la forme :

$$\sum_{k=0}^{n} (-1)^k \binom{n}{k}^3 = \begin{cases} (-1)^p \binom{3p}{p,p,p} & \text{si } n = 2p, \\ 0 & \text{si } n = 2p+1. \end{cases}$$

Pour $S = \{0,1,2,3\}^2$, la méthode de Fasenmyer renvoie (rapidement) un résultat positif, conduisant à la relation

$$3(3n-2)(3n-4)F_{n-2} + n^2 F_n = 0.$$

L'équation obtenue est :

$$\begin{aligned}
&n^2(3n-5)f_{n,k} + (9n^3 - 24n^2 + 17n - 4)f_{n-1,k-1} \\
&+ (3n-4)(21n^2 - 49n + 24)f_{n-2,k-1} + (3n-4)(3n^2 - 7n + 3)f_{n-2,k-2} \\
&+ 3(3n-2)(n-2)^2 f_{n-3,k-1} - 3(3n-2)(n-2)^2 f_{n-3,k-2} \\
&+ (3n-2)(n-2)^2 f_{n-3,k-3} + (-9n^3 + 24n^2 - 17n + 4)f_{n-1,k} \\
&+ (3n-4)(3n^2 - 7n + 3)f_{n-2,k} - (3n-2)(n-2)^2 f_{n-3,k} = 0.
\end{aligned}$$

Autres exemples typiques par cette méthode

$$\sum_{i,j} \binom{i+j}{i}^2 \binom{4n-2i-2j}{2n-2i} = (2n+1)\binom{2n}{n}^2 \qquad \text{(Andrews–Paule)}$$

$$\sum_{j,k} (-1)^{j+k} \binom{j+k}{k+l} \binom{r}{j} \binom{n}{k} \binom{s+n-j-k}{m-j}$$

$$= (-1)^l \binom{n+r}{n+l} \binom{s-r}{m-n-l}$$

(Graham–Knuth–Patashnik)

$$\sum_{r,s} (-a)^{n+r+s} \binom{n}{r} \binom{n}{s} \binom{n+s}{s} \binom{n+r}{r} \binom{2n-r-s}{n} = \sum_k \binom{n}{k}^4$$

(Petkovšek–Wilf–Zeilberger)

Algorithme de Gosper

ENTRÉE : terme hypergéométrique f_k.
SORTIE : fraction rationnelle $R(k)$ telle que Rf soit une somme indéfinie en k de f, ou une preuve qu'aucune telle R n'existe.

VISION SIMPLIFIÉE DE L'ALGORITHME : L'équation

$$f_k = R(k+1)f_{k+1} - R(k)f_k$$

en une fraction rationnelle R s'écrit encore

$$1 = R(k+1)\rho(k) - R(k),$$

équation qui se résout par l'algorithme de décision d'Abramov : si aucune R n'est trouvée, on a une preuve qu'aucune n'existe.

Algorithme original :

1. Calculer le rapport $\rho(k) = f_{k+1}/f_k$ et le représenter sous la forme (unique)
$$\rho(k) = \frac{p(k+1)}{p(k)} \frac{q(k)}{r(k+1)}$$
de façon que $\mathrm{pgcd}(p, q) = \mathrm{pgcd}(p, r) = \mathrm{pgcd}\big(q(k), r(k+h)\big) = 1$ pour tout entier $h > 0$.

2. Le changement de variable $R = rS/p$ ramène l'équation en la fraction rationnelle R en une équation en un polynôme S :
$$p(k) = q(k)S(k+1) - r(k)S(k).$$

3. Résoudre (des bornes sur le degré existent). Si un S est trouvé, renvoyer $R = rS/p$; sinon, on a une preuve qu'aucune R n'existe.

Exemple d'exécution de l'algorithme de Gosper

$$\sum_{k=0}^{n-1} \boxed{\frac{(-1)^k(4k+1)\binom{2k+1}{k}}{4^k(4k^2-1)}} = -\frac{2(n+1)}{4n+1} \boxed{\frac{(-1)^n(4n+1)\binom{2n+1}{n}}{4^n(4n^2-1)}} - 2.$$

Le sommant hypergéométrique est donné par :

$$\rho = -\frac{1}{2}\frac{(4k+5)(2k-1)}{(4k+1)(k+2)},$$

$$p(k) = 4k+1, \qquad q(k) = \frac{1}{2} - k, \qquad r(k) = k+2.$$

Algorithme de Gosper paramétré

ENTRÉE : un terme hypergéométrique f_k et des fractions rationnelles $s_0(k), \ldots, s_m(k)$.
SORTIE : une fraction rationnelle $R(k)$ et des constantes $\eta_0, \ldots, \eta_m$ telles que Rf soit une somme indéfinie en k de

$$\big(\eta_0 s_0(k) + \cdots + \eta_m s_m(k)\big) f_k,$$

ou une preuve qu'aucune telle R n'existe pour aucune famille $\{\eta_i\}$.

IDÉE DE L'ALGORITHME : Les paramètres η_i n'interviennent que de façon linéaire et au second membre de l'équation polynomiale en $S(k)$. Le calcul de S se résume à la résolution d'un système linéaire lors de laquelle on peut prendre en compte les inconnues additionnelles $\eta_0, \ldots, \eta_m$.

Algorithme de Zeilberger

ENTRÉE : terme hypergéométrique $f_{n,k}$.
SORTIE : fractions rationnelles $\eta_0(n), \ldots, \eta_d(n), \phi(n,k)$ pour d minimal telles que

$$\eta_d(n) f_{n+d,k} + \ldots \eta_0(n) f_{n,k} = \phi(n, k+1) f_{n,k+1} - \phi(n,k) f_{n,k}.$$

Terminaison non garantie en général ; l'est dans le cas « holonome ».
Critère explicite récent dû à Abramov.

ALGORITHME : Calculer les fractions rationnelles $s_i(n,k)$ telles que $f_{n+i,k} = s_i(n,k) f_{n,k}$ et utiliser l'algorithme de Gosper paramétré.

Exemple d'exécution de l'algorithme de Zeilberger

Les polynômes orthogonaux de Jacobi $P_n^{(\alpha,\beta)}(x)$ s'expriment en termes de la fonction hypergéométrique de Gauss ${}_2F_1\left(\begin{smallmatrix} a,b \\ c \end{smallmatrix}\middle| z\right)$:

$$P_n^{(\alpha,\beta)}(x) = \frac{(\alpha+1)_n}{n!}\,{}_2F_1\left(\begin{matrix} -n, n+\alpha+\beta+1 \\ \alpha+1 \end{matrix}\middle|\frac{1-x}{2}\right),$$

pour

$${}_2F_1\left(\begin{matrix} a, b \\ c \end{matrix}\middle| z\right) = \sum_{k=0}^{\infty} \frac{(a)_k(b)_k}{(c)_k k!} z^k$$

où

$$(u)_k = u(u+1)\cdots(u+k-1).$$

L'algorithme fournit la récurrence en n sur $P_n^{(\alpha,\beta)}(x)$,

$$\begin{aligned}0 &= 2(n+2)(n+\alpha+\beta+2)(2n+\alpha+\beta+2)P_{n+2}^{(\alpha,\beta)}(x)\\ &\quad - \big((2n+\alpha+\beta+2)_3 x + (2n+\alpha+\beta+3)(\alpha-\beta)(\alpha+\beta)\big)P_{n+1}^{(\alpha,\beta)}(x)\\ &\qquad + 2(n+\alpha+1)(n+\beta+1)(2n+\alpha+\beta+4)P_n^{(\alpha,\beta)}(x).\end{aligned}$$

Il fournirait aussi des relations de récurrence en α ou en β (relations de contiguïté).

Le sommant est

$$f_k = \frac{(\alpha+1)_n}{n!} \frac{(-n)_k (n+\alpha+\beta+1)_k}{(\alpha+1)_k k!} \frac{(1-x)^k}{2^k}.$$

Le rapport f_{k+1}/f_k est

$$\rho = \frac{1}{2} \frac{(n-k)(n+\alpha+\beta+1+k)(-1+x)}{(k+1)(\alpha+1+k)}.$$

La récurrence paramétrée à résoudre en solutions rationnelles est

$$
\begin{aligned}
&(n+2-k)(n+1-k)(n-k)(n+2+\alpha+\beta) \\
&\qquad (n+\alpha+\beta+1)(n+\alpha+\beta+1+k)(-1+x)R(k+1) \\
&- 2(k+1)(\alpha+1+k)(n+\alpha+\beta+1) \\
&\qquad (n+1-k)(n+2-k)(n+2+\alpha+\beta)R(k) \\
&= 2\eta_0(k+1)(\alpha+1+k)(n+\alpha+\beta+1) \\
&\qquad (n+1-k)(n+2-k)(n+2+\alpha+\beta) \\
&+ 2\eta_1(n+\alpha+\beta+1+k)(\alpha+1+n)(k+1) \\
&\qquad (\alpha+1+k)(n+2-k)(n+2+\alpha+\beta) \\
&+ 2\eta_2(\alpha+2+n)(\alpha+1+n)(n+\alpha+\beta+2+k) \\
&\qquad (n+\alpha+\beta+1+k)(k+1)(\alpha+1+k).
\end{aligned}
$$

Algorithme d'Abramov

Entrée : récurrence linéaire de la forme
$a_d(k)f_{k+d} + \cdots + a_0(k)f_k = b(k)$ pour des polynômes a_i et b.
Sortie : solutions rationnelles de la récurrence, ou une preuve qu'il n'en existe pas.

Algorithme :

1. calculer le plus grand entier positif N tel que $a_0(k+N)$ et $a_d(k-d)$ ont un p. g. c. d. non trivial ;
2. si aucun tel N n'existe, poser $v = 1$, sinon poser
$$v = \text{p. g. c. d.}\left(\prod_{i=0}^{N} a_0(k+i), \prod_{i=0}^{N} a_d(k-d-i)\right) \;;$$

3. après avoir posé $f_k = u(k)/v(k)$, on est ramené à rechercher les solutions polynomiales $u(k)$ d'une nouvelle équation de récurrence linéaire de même ordre d ;
4. on sait borner explicitement le degré d'une solution polynomiale, puis on résoud, par exemple par coefficients indéterminés.

Remarque : très souvent, il suffit d'injecter λx^δ dans une récurrence linéaire pour trouver le degré de ses solutions polynomiales.

Retour à l'exemple des polynômes de Jacobi

Un multiple du dénominateur de toute solution sous forme réduite est

$$(n+2-k)(n+1-k).$$

La récurrence prend la nouvelle forme

$$\begin{aligned}\big(-(n+2+\alpha+\beta)(n+\alpha+\beta+1)(-1+x)k^2+\cdots\big)S(k+1)\\ +\big(-2(n+2+\alpha+\beta)(n+\alpha+\beta+1)k^2+\cdots\big)S(k)\\ =(c_0k^4+\cdots)\eta_0+(c_1k^4+\cdots)\eta_1+(c_2k^4+\cdots)\eta_2.\end{aligned}$$

Ainsi, toute solution $S(k)$ est de degré au plus 2 en k.

Après résolution, on obtient $P(n,k) = Q(n,k+1) - Q(n,k)$ pour

$$\begin{aligned}
P(n,k) &= 2(\beta+n+1)(\alpha+1+n)(\alpha+2n+4+\beta)u(n,k) \\
&\quad - (3+\alpha+\beta+2n)(\alpha^2+\alpha^2 x+4\alpha x n+2\alpha\beta x+6\alpha x+12xn \\
&\quad + \beta^2 x+4xn\beta+4xn^2+6\beta x+8x-\beta^2)u(n+1,k) \\
&\quad + 2(n+2)(n+2+\alpha+\beta)(\alpha+2+2n+\beta)u(n+2,k),
\end{aligned}$$

$$Q(n,k) = -2k(3+\alpha+\beta+2n)(\alpha+2+2n+\beta)(\alpha+1+n)(\alpha+k)$$
$$(\alpha+2n+4+\beta)u(n,k)/(n+1-k)/(n+2-k)/(n+\alpha+\beta+1).$$

D'où la récurrence annoncée par *creative telescoping*.

Suites ∂-finies

Généralisation des suites polynomialement récursives à un indice.

Un terme $u_{n,k}$ est dit ∂-fini lorsque l'ensemble de ses décalés $u_{n+i,k+j}$ engendre un espace vectoriel de dimension finie sur $\mathbb{C}(n,k)$.

Un terme $u_k(z)$ est dit ∂-fini lorsque l'ensemble de ses dérivées et décalés $u_{k+j}^{(i)}(z)$ engendre un espace vectoriel de dimension finie sur $\mathbb{C}(k,z)$.

Exemple : familles de polynômes orthogonaux classiques, fonctions de Bessel, fonctions hypergéométriques et généralisations, sommes indéfinies de suites hypergéométriques, etc, l'algèbre de ces familles.

Algorithme de Gosper étendu

Entrée : un espace vectoriel V de base $\big(b_1(k), \dots, b_r(k)\big)$ stable par décalage en k ; un terme ∂-fini f_k de V.

Sortie : des fractions rationnelles $\phi_1(k)$, ..., $\phi_r(k)$ telles que $\phi_1 b_1 + \cdots + \phi_r b_r$ soit une somme indéfinie en k de f, ou une preuve que de telles ϕ_i n'existent pas.

Algorithme :

1. Poser $F_k = \phi_1(k)b_1(k) + \cdots + \phi_r(k)b_r(k)$ pour $\phi_i \in \mathbb{C}(k)$.
2. Extraire les coefficients de $F_{k+1} - F_k - f_k$ sur les b_i et obtenir un système de relations de récurrences du premier ordre en les ϕ_i.
3. Résoudre en les solutions rationnelles (algorithme de décision).
4. Si des ϕ_i sont trouvées, les renvoyer ; sinon, on a une preuve que de telles ϕ_i n'existent pas.

Exemples typiques par cette méthode

$$\sum_{k=0}^{n}(\pm 1)^k(k+\alpha)C_k^{(\alpha)}(x)$$

$$= \frac{(\pm 1)^n}{2(1\mp x)}\Big((n+2\alpha)C_n^{(\alpha)}(x) \mp (n+1)C_{n+1}^{(\alpha)}(x)\Big)$$

$$(C_k^{(\alpha)}(x) = \text{polynômes ultrasphériques}),$$

$$\sum_{k=1}^{n}\frac{(-1)^k}{\binom{m}{k}}H_k = \frac{(-1)^n}{\binom{m}{n}}\left(\frac{n+1}{m+2}H_n + \frac{m+1-n}{(m+2)^2}\right) - \frac{m+1}{(m+2)^2},$$

$$\sum_{k=1}^{n}H_k^3 = (n+1)H_n^3 - \frac{3}{2}(2n+1)H_n^2 + 3(2n+1)H_n + \frac{1}{2}H_n^{(2)} - 6n$$

$$\big(H_n^{(2)} = 1 + \frac{1}{4} + \frac{1}{9} + \cdots + \frac{1}{n^2}\big).$$

Algorithme de Zeilberger étendu

Entrée : un espace vectoriel V de base $\big(b_1(n,k),\dots,b_r(n,k)\big)$ stable par les décalages en n et k ; un terme ∂-fini $f_{n,k}$ de V.
Sortie : des fractions rationnelles $\eta_0(n)$, ..., $\eta_d(n)$, $\phi_1(n,k)$, ..., $\phi_r(n,k)$ pour d minimal telles que

$$\eta_d(n)f_{n+d,k} + \dots \eta_0(n)f_{n,k} = g_{n,k+1} - g_{n,k},$$

pour

$$g_{n,k} = \phi_1(n,k)b_1(n,k) + \dots + \phi_r(n,k)b_r(n,k).$$

Terminaison non garantie en général ; l'est dans le cas « holonome ».

Algorithme : Se ramener à une variante paramétrée de l'algorithme de Gosper étendu.

Exemples typiques par cette méthode

$$\sum_{n=0}^{\infty} J_{2n+1/2}(x) = \int_0^x \frac{\cos t}{\sqrt{2\pi t}}\, dt$$

$$\sum_{k=0}^{n} \binom{n}{k}^2 \binom{n+k}{k}^2 = \sum_{k=0}^{n} \binom{n}{k}\binom{n+k}{k} \sum_{j=0}^{k} \binom{k}{j}^3 \quad \text{(van der Poorten)}$$

$$\sum_{k=0}^{n} \left(\sum_{j=0}^{k} \binom{n}{j} \right)^3 = \frac{n}{2} 8^n + 8^n - \frac{3n}{4} 2^n \binom{2n}{n} \qquad \text{(Calkin)}$$

$$\sum_{k=0}^{n} \sum_{l=0}^{n+1-k} \frac{l}{n+1} \binom{n+1}{k} \binom{n+1}{k+l} = \binom{2n}{n} \quad \text{(Essam et Guttmann)}$$

Non minimalité des opérateurs calculés

$$\sum_{k=0}^{n}(-1)^k\binom{n}{k}\binom{pk}{n} = (-p)^n \qquad \text{(ordre } p-1 \text{ au lieu de 1),}$$

$$\sum_{k=0}^{n}\frac{1}{pk+1}\binom{pk+1}{k}\binom{p(n-k)}{n-k} = \binom{pn+1}{n}$$

(ordre 2 au lieu de 1 pour $p = 3$).

$$\eta_d(n)u_{n+d,k} + \cdots + \eta_0(n)u_{n,k} = v_{n,k+1} - v_{n,k}$$

L'existence de v est une contrainte sur d !

Cette relation ne tient pas compte du lieu de sommation !