

## Sommation hypergéométrique<sup>1</sup>

### Résumé

Les suites hypergéométriques sont très courantes dans les applications. Leur algorithmique fait partie des succès du calcul formel. Les deux problèmes principaux sont la sommation indéfinie et la sommation définie. Les algorithmes correspondants sont dus à Gosper et Zeilberger.

Dans tout ce qui suit,  $\mathbb{K}$  est un corps de caractéristique nulle.

### 1. Sommation indéfinie

**1.1. Problème de la sommation indéfinie.** La sommation indéfinie est l'analogue discret du calcul de primitives.

DÉFINITION 1. *Étant donnée une suite  $(f_n) \in \mathbb{K}^{\mathbb{N}}$ , on appelle  $(F_n) \in \mathbb{K}^{\mathbb{N}}$  une somme indéfinie de  $(f_n)$  si*

$$\forall n \in \mathbb{N}, \quad F_{n+1} - F_n = f_n.$$

Le lien entre sommation indéfinie et somme est le même qu'entre primitive et intégrale : si  $m \leq p \in \mathbb{N}$ ,

$$\sum_{n=m}^p f_n = \sum_{n=m}^p (F_{n+1} - F_n) = F_{p+1} - F_m.$$

### 1.2. Sommation indéfinie hypergéométrique.

DÉFINITION 2. *Une suite  $(u_n) \in \mathbb{K}^{\mathbb{N}}$  est dite hypergéométrique sur  $\mathbb{K}$  s'il existe une fraction rationnelle  $r(n) = p(n)/q(n) \in \mathbb{K}(n)$  telle que*

$$\forall n \in \mathbb{N}, \quad q(n)u_{n+1} = p(n)u_n.$$

EXEMPLE 1. – Une suite géométrique est hypergéométrique : prendre  $r(n) = \alpha$  où  $\alpha$  est la raison de la suite. En ce sens, les suites hypergéométriques sont une généralisation des suites géométriques.  
– La suite factorielle est hypergéométrique :

$$\forall n \in \mathbb{N}, \quad \frac{(n+1)!}{n!} = n+1.$$

–  $k \in \mathbb{N}$  étant fixé, les coefficients binomiaux  $\binom{n}{k}$  forment une suite hypergéométrique :

$$\forall n \in \mathbb{N}, \quad \frac{\binom{n+1}{k}}{\binom{n}{k}} = \frac{\frac{(n+1)!}{k!(n+1-k)!}}{\frac{n!}{k!(n-k)!}} = \frac{n+1}{n+1-k}.$$

<sup>1</sup>La première rédaction de ce chapitre est due à Aurel Page.

- $n \in \mathbb{N}$  étant fixé, les coefficients binomiaux  $\binom{n}{k}$  forment une suite hypergéométrique :

$$\forall k \in \mathbb{N}, \quad \frac{\binom{n}{k+1}}{\binom{n}{k}} = \frac{\frac{n!}{(k+1)!(n-k-1)!}}{\frac{n!}{k!(n-k)!}} = \frac{n-k}{k+1}.$$

- Le cas général de suites hypergéométriques sur  $\mathbb{C}$  s'écrit

$$u_n = CA^n \frac{\prod_{i=1}^p (a_i)(a_i+1) \cdots (a_i+n)}{\prod_{j=1}^q (b_j)(b_j+1) \cdots (b_j+n)},$$

avec  $C, A, a_1, \dots, a_p, b_1, \dots, b_q$  dans  $\mathbb{C}$ . Il est clair que cette suite est hypergéométrique, et réciproquement, toute fraction rationnelle peut s'écrire  $AP(n)/Q(n)$  où  $p$  et  $q$  sont unitaires et on obtient la formule ci-dessus en nommant  $a_1, \dots, a_p$  les racines de  $P$  et  $b_1, \dots, b_q$  celles de  $Q$ .

PROBLÈME 1. On se restreint aux suites hypergéométriques.

- **Entrée** :  $r(n) \in \mathbb{K}(n)$ , et sous-entendue  $(u_n) \in \mathbb{K}^{\mathbb{N}}$  telle que  $\forall n \in \mathbb{N}$ ,  $u_{n+1}/u_n = r(n)$ .
- **Sortie** : une suite hypergéométrique  $(v_n)$  telle que  $\forall n \in \mathbb{N}$ ,  $v_{n+1} - v_n = u_n$ , ou FAIL si une telle suite n'existe pas. Lorsqu'elle existe, on dit que  $(v_n)$  est une somme hypergéométrique de  $(u_n)$ .

### 1.3. Algorithme de Gosper.

LEMME 1. Soit  $(u_n) \in \mathbb{K}^{\mathbb{N}}$  une suite hypergéométrique. Si  $(u_n)$  admet une somme hypergéométrique  $(v_n)$ , alors il existe une fraction rationnelle  $t(n) \in \mathbb{K}(n)$  telle que  $\forall n \in \mathbb{N}$ ,  $v_n = t(n)u_n$ .

DÉMONSTRATION. Soit  $s(n) \in \mathbb{K}(n)$  telle que  $v_{n+1}/v_n = s(n)$ . Alors :

$$\forall n \in \mathbb{N}, \quad v_{n+1} - v_n = (s(n) - 1)v_n = u_n$$

D'où le résultat en prenant  $t(n) = 1/(s(n) - 1)$ . □

REMARQUE 1. On obtient même l'équation que doit vérifier  $t(n)$  :

$$\forall n \in \mathbb{N}, \quad t(n+1)r(n)u_n - t(n)u_n = u_n$$

d'où

$$t(n+1)r(n) - t(n) = 1$$

(équation d'inconnue  $t(n)$  rationnelle).

IDÉE 1 (Gosper, 1978). On se débarrasse des différences entières entre racines et pôles en écrivant  $r(n)$  sous la forme (appelée forme de Gosper)

$$(1) \quad r(n) = \frac{a(n)}{b(n)} \frac{c(n+1)}{c(n)}, \quad a, b, c \in \mathbb{K}[n]$$

avec  $\forall k \in \mathbb{N}$ ,  $\text{pgcd}(a(n), b(n+k)) = 1$ , puis on cherche  $t(n)$  sous la forme  $t(n) = b(n-1)x(n)/c(n)$ .

L'équation sur  $t(n)$  devient

$$\frac{b(n)}{c(n+1)}x(n+1) - \frac{a(n)}{b(n)} \frac{c(n+1)}{c(n)} - \frac{b(n-1)}{c(n)}x(n) = 1$$

soit finalement

$$(2) \quad a(n)x(n+1) - b(n-1)x(n) = c(n).$$

**THÉORÈME 11** (Gosper 1978). *Avec les notations précédentes, si  $x(n)$  est une fraction rationnelle solution de (2), alors c'est un polynôme.*

**DÉMONSTRATION.** On écrit  $x(n) = p(n)/q(n)$  avec  $p, q \in \mathbb{K}[n]$  premiers entre eux, et on injecte dans l'équation :

$$a(n)p(n+1)q(n) - b(n-1)q(n+1)p(n) = q(n)q(n+1)c(n).$$

On a donc  $q(n)|b(n-1)q(n+1)p(n)$  d'où  $q(n)|b(n-1)q(n+1)$ . De même  $q(n+1)|a(n)q(n)$ . En itérant, pour tout  $K \in \mathbb{N}^*$  on déduit

$$q(n)|b(n-1)b(n)\cdots b(n+K-2)q(n+K), \quad q(n)|a(n-1)a(n-2)\cdots a(n-K)q(n-K).$$

En choisissant  $K$  assez grand pour que  $\text{pgcd}(q(n), q(n+K)) = 1$ , on en déduit

$$q(n)|\text{pgcd}(b(n-1)b(n)\cdots b(n+K-2), a(n-1)a(n-2)\cdots a(n-K)) = 1,$$

où la dernière égalité est conséquence des hypothèses sur  $a$  et  $b$ .  $\square$

Le principe de l'algorithme de Gosper est donné en Algorithme 1. Il dépend de deux autres algorithmes, pour le calcul de la forme de Gosper et pour la recherche de solutions polynomiales.

---

#### Algorithme 1 Algorithme de Gosper

---

**Entrées:**  $r(n) \in \mathbb{K}(n)$  telle que  $\forall n \in \mathbb{N}, u_{n+1}/u_n = r(n)$ .

**Sorties:**  $f(n)$  telle que la suite hypergéométrique définie par  $\forall n \in \mathbb{N}, v_n = f(n)u_n$  vérifie  $\forall n \in \mathbb{N}, v_{n+1} - v_n = u_n$ , ou FAIL si  $(u_n)$  n'admet pas de somme hypergéométrique.

- 1: Calculer une forme de Gosper de  $r(n)$ .
  - 2: Trouver une solution polynomiale  $x(n) \in \mathbb{K}[n]$  de (2).
  - 3: **si** il n'y a pas de solution **alors**
  - 4:   **renvoyer** FAIL
  - 5: **sinon**
  - 6:   **renvoyer**  $\frac{b(n-1)}{c(n)}x(n)$
  - 7: **fin si**
- 

---

#### Algorithme 2 Forme de Gosper

---

**Entrées:**  $r(n) = P(n)/Q(n)$ ,  $\text{pgcd}(P, Q) = 1$

**Sorties:**  $a, b, c \in \mathbb{K}[n]$  obéissant à (1) et tels que  $\forall k \in \mathbb{N}, \text{pgcd}(a(n), b(n+k)) = 1$ .

- 1:  $R(k) := \text{Res}_n(P(n), Q(n+k))$
  - 2: Calculer  $0 < h_1 < h_2 < \cdots < h_N$  les racines entières de  $R$ .
  - 3:  $a(n) := P(n); b(n) := Q(n); c(n) := 1$
  - 4: **pour**  $i$  allant de 1 à  $N$  **faire**
  - 5:    $g_i := \text{pgcd}(a(n), b(n+h_i))$
  - 6:    $a(n) := a(n)/g_i(n)$
  - 7:    $b(n) := b(n)/g_i(n)$
  - 8:    $c(n) := c(n)g_i(n-1)g_i(n-2)\cdots g_i(n-h_i)$
  - 9: **fin pour**
  - 10: **renvoyer**  $a, b, c$ .
-

Le calcul de la forme de Gosper est donné en Algorithme 2. Dans un premier temps, il calcule l'ensemble des différences entières possibles entre les racines du numérateur et du dénominateur. Dans un second temps (la boucle), ces différences sont accumulées dans le polynôme  $c$ . La correction de l'algorithme provient d'un invariant simple à observer par récurrence : à chaque itération, (1) est satisfaite, et après l'étape  $i$ ,  $\text{pgcd}(a(n), b(n + h_i)) = 1$ . Ces deux propriétés restent donc vraies à la fin de l'algorithme.

La recherche de solutions polynomiales de l'équation (2) procède aussi en deux temps.

D'abord, on va chercher une borne sur le degré de  $x$ , ensuite on peut résoudre par coefficients indéterminés (le système est même linéaire). La discussion sur le degré distingue deux cas :

1. Cas facile :  $\deg a \neq \deg b$ , ou bien  $\deg a = \deg b$  et  $\text{lc}(a) \neq \text{lc}(b)$ <sup>2</sup>. Alors le degré du membre gauche de (2) est donné par celui d'un de ses deux termes et on en déduit  $\deg x \leq \deg c - \max(\deg a, \deg b)$ .
2. Cas restant : on note  $a(n) = \lambda n^d + \alpha n^{d-1} + \dots$ ,  $b(n-1) = \lambda n^d + \beta n^{d-1} + \dots$ ,  $x(n) = \gamma n^D + \dots$  où "... " désigne des éléments de degré inférieur à celui des premiers termes,  $\lambda \neq 0, \gamma \neq 0$  (soit  $d = \deg a = \deg b$  et  $D = \deg x$ ). Alors on peut réécrire l'équation :

$$\underbrace{a(n)}_{\lambda n^d + \dots} \underbrace{(x(n+1) - x(n))}_{\gamma D n^{D-1} + \dots} + \underbrace{(a(n) - b(n-1))}_{(\alpha - \beta)n^{d-1}} \underbrace{x(n)}_{\gamma n^D + \dots} = c(n)$$

qui donne

$$(\lambda \gamma D + (\alpha - \beta)\gamma)n^{d+D-1} + \dots = c(n)$$

Alors, soit  $D \leq \deg c - d + 1$ , soit  $\lambda \gamma D + (\alpha - \beta)\gamma = 0$ , d'où  $D = (\beta - \alpha)/\lambda$  (car  $\gamma \neq 0$ ). On obtient donc dans ce cas  $D \leq \max(\deg c - d + 1, \frac{\beta - \alpha}{\lambda})$ .

Une fois le degré de  $x$  borné, on peut résoudre l'équation par coefficients indéterminés.

## 2. Sommation définie

Dans toute cette partie on suppose de plus que  $\mathbb{K}$  est un corps topologique.

### 2.1. Problème de la sommation définie.

DÉFINITION 3. On dit qu'une suite  $(F_{n,k}) \in \mathbb{K}^{\mathbb{N} \times \mathbb{N}}$  est hypergéométrique si  $F_{n+1,k}/F_{n,k}$  et  $F_{n,k+1}/F_{n,k}$  sont des fractions rationnelles dans  $\mathbb{K}(n, k)$ .

PROBLÈME 2. Étant donné une suite hypergéométrique  $(F_{n,k}) \in \mathbb{K}^{\mathbb{N} \times \mathbb{N}}$ , on cherche une récurrence linéaire à coefficients polynomiaux pour la suite  $(u_n)$  définie par

$$\forall n \in \mathbb{N}, \quad u_n = \sum_{k \in \mathbb{N}} F_{n,k}$$

EXEMPLE 2. Des identités typiques de ce qui est calculable dans ce contexte sont :

$$\sum_{k=0}^n \binom{n}{k} = 2^n, \quad \sum_{k=0}^n \binom{n}{k}^2 = \binom{2n}{n}.$$

<sup>2</sup>Si  $P \in \mathbb{K}[n]$ , on note  $\text{lc}(P)$  son coefficient dominant (*leading coefficient*).

## 2.2. Algorithme de Zeilberger.

IDÉE 2. (Zeilberger) On cherche des polynômes  $t_0(n), \dots, t_p(n) \in \mathbb{K}[n]$  et  $G_{n,k}$  hypergéométrique tels que

$$(3) \quad t_0(n)F_{n,k} + t_1(n)F_{n+1,k} + \dots + t_p(n)F_{n+p,k} = G_{n,k+1} - G_{n,k}.$$

Cela permet de trouver une solution du problème en sommant sur  $k$  : si  $\lim_{k \rightarrow \pm\infty} G_{n,k} = 0$ , alors

$$t_0(n)u_n + t_1(n)u_{n+1} + \dots + t_p(n)u_{n+p} = 0.$$

L'algorithme de Zeilberger part de l'observation que  $G_{n,k}$  est une somme indéfinie en  $k$  d'une suite hypergéométrique. En effet, avec  $g_{n,k}$  le membre gauche de (3), on a

$$\frac{g_{n,k+1}}{g_{n,k}} = \frac{\sum_i t_i(n)F_{n+i,k+1}}{\sum_i t_i(n)F_{n+i,k}}$$

qui se récrit

$$\frac{g_{n,k+1}}{g_{n,k}} = \frac{\sum_{i=0}^p t_i(n) \frac{F_{n+i,k+1}}{F_{n,k+1}} F_{n,k+1}}{\sum_{i=0}^p t_i(n) \frac{F_{n+i,k}}{F_{n,k}} F_{n,k}} \in \mathbb{K}(n, k).$$

L'idée est alors d'utiliser l'algorithme de Gosper dans le corps  $\mathbb{K}(n)$ , avec des  $t_i$  indéterminés.

1. Posons  $P_n(k) = \sum_{i=0}^p t_i(n)F_{n+i,k}/F_{n,k}$ ,  $Q_n(k) = F_{n,k+1}/F_{n,k}$  et  $R_n(k) = g_{n,k+1}/g_{n,k}$ . Alors  $P_n$  dépend linéairement des  $t_i$ ,  $Q_n$  n'en dépend pas, et on a

$$R_n(k) = \frac{P_n(k+1)}{P_n(k)} Q_n(k)$$

Ainsi, en mettant  $Q_n$  sous forme de Gosper

$$Q_n(k) = \frac{A_n(k)}{B_n(k)} \frac{c_n(k+1)}{c_n(k)}$$

et en posant  $C_n(k) = P_n(k)c_n(k)$ , on obtient une forme de Gosper pour  $R_n$

$$R_n(k) = \frac{A_n(k)}{B_n(k)} \frac{C_n(k+1)}{C_n(k)}$$

où  $A_n$  et  $B_n$  ne dépendent pas des  $t_i$ , et  $C_n$  en dépend linéairement.

2. L'équation qu'il faut alors résoudre est

$$A_n(k)X(k+1) - B_n(k-1)X(k) = C_n(k)$$

Dans cette équation, seul  $C_n$  dépend des  $t_i$ , et de façon linéaire. De plus, dans l'algorithme de Gosper, les bornes sur le degré du polynôme inconnu dépendent uniquement de  $A_n$  et de  $B_n$ , qui ne dépendent pas des  $t_i$ , et du degré en  $k$  de  $C_n$  qui peut être borné indépendamment des  $t_i$ . Cette équation est donc un système linéaire fini, d'inconnues à la fois les coefficients de  $X$  et les  $t_i$ , qu'on peut résoudre.

3. Il suffit alors d'essayer des ordres  $p$  successivement dans l'espoir de trouver une récurrence.

REMARQUE 2. Un théorème de Wilf & Zeilberger assure l'existence d'une telle récurrence (et donc la terminaison de l'algorithme) pour une sous-classe des suites hypergéométriques appelées "proprement" hypergéométriques, c'est-à-dire celles qui peuvent s'écrire sous la forme

$$P(n, k)A^k \frac{\prod_{i=1}^p (a_i n + b_i k + c_i)!}{\prod_{j=1}^q (u_j n + v_j k + w_j)!},$$

où  $P$  est un polynôme, les  $a_i, b_i, u_j, v_j$  sont des entiers, et  $p$  et  $q$  sont des entiers positifs ou nuls.