Ускоренное неопределенное суммирование

Зима Е.В.
Wilfrid Laurier University
Waterloo, Canada
ezima@wlu.ca

Обозначения:

$\mathbb{K}$ – поле характеристики 0,

x – независимая переменная,

E – оператор сдвига по переменной x ($Ef(x) = f(x+1)$),

Проблема неопределенного суммирования:

по данному выражению $F(x)$ найти $G(x)$, являющуюся решением

$$(E-1)G(x) = F(x). \tag{1}$$

$$G(x) = \sum_x F(x) + c$$

Если у (1) решения не существует, рассматривается проблема аддитивной декомпозиции:
по данному $F(x)$ найти такие $R(x), H(x)$, что

$$F(x) = (E - 1)R(x) + H(x), \tag{2}$$

и $H(x)$ в некотором смысле проще, чем $F(x)$. Если $G(x)$, удовлетворяющее (1) существует, пара $R(x) = G(x)$ и $H(x) = 0$ рассматривается как решение проблемы аддитивной декомпозиции.

$$F(x) = R(x) + \sum_x H(x)$$

Список литературы

[1] S. A. Abramov. The summation of rational functions. *Ž. Vyčisl. Mat. i Mat. Fiz.*, 11:1071–1075, 1971.

[2] S. A. Abramov. The rational component of the solution of a first order linear recurrence relation with rational right hand side. *Ž. Vyčisl. Mat. i Mat. Fiz.*, 15(4):1035–1039, 1090, 1975.

[3] S. A. Abramov. Indefinite sums of rational functions. In *Proceedings of the 1995 International Symposium on Symbolic and Algebraic Computation*, ISSAC '95, pages 303–308, New York, NY, USA, 1995. ACM.

[4] S. A. Abramov, M. Bronstein, and M. Petkovšek. On polynomial solutions of linear operator equations. In *Proceedings of the 1995 International Symposium on Symbolic and Algebraic Computation*, ISSAC '95, pages 290–296, New

York, NY, USA, 1995. ACM.

[5] S. A. Abramov and M. Petkovšek. Rational normal forms and minimal decompositions of hypergeometric terms. *Journal of Symbolic Computation*, 33(5):521–543, 2002. Computer algebra (London, ON, 2001).

[6] O. Bachmann, P. S. Wang, and E. V. Zima. Chains of recurrences — a method to expedite the evaluation of closed-form functions. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, ISSAC '94, pages 242–249, New York, NY, USA, 1994. ACM.

[7] G. Boole. *A Treatise on the Calculus of Finite Differences*. Cambridge Library Collection. Cambridge University Press, Cambridge, 2009. Reprint of the 1860 original.

[8] A. Bostan, B. Salvy, and E. Schost. Power series composition and change of basis. In *Proceedings of the International*

Symposium on Symbolic and Algebraic Computation, ISSAC '08, pages 269–276, New York, NY, USA, 2008. ACM.

[9] Euler. *Foundations of Differential Calculus*. Springer-Verlag, New York, 2000. Translated from the Latin by John D. Blanton.

[10] J. V. Z. Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.

[11] J. Gerhard. *Modular Algorithms in Symbolic Summation and Symbolic Integration*, volume 3218 of *Lecture Notes in Computer Science*. Springer, 2004.

[12] J. Gerhard, M. Giesbrecht, A. Storjohann, and E. V. Zima. Shiftless decomposition and polynomial-time rational summation. In *Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation*, pages

119–126 (electronic), New York, 2003. ACM.

[13] C. Jordan. *Calculus of Finite Differences.* Hungarian Agent Eggenberger Book-Shop, Budapest, 1939.

[14] C. Jordan. *Calculus of Finite Differences.* Third Edition. Introduction by Harry C. Carver. Chelsea Publishing Co., New York, 1965.

[15] M. Karr. Summation in finite terms (preliminary version). Technical Report CA-7602-2911, Massachusetts Computer Associates Inc., 1976.

[16] J. C. Lafon. Summation in finite terms. In B. Buchberger, G. E. Collins, R. Loos, and R. Albrecht, editors, *Computer algebra. Symbolic and algebraic computation*, pages 71–77. Springer, Vienna, 1983.

[17] Y.-K. Man. On computing closed forms for indefinite summations. *Journal of Symbolic Computation*, 16(4):355–376,

Oct. 1993.

[18] R. Moenck. On computing closed forms for summations. In *Proceedings of the 1977 MACSYMA Users' Conference*, pages 225–236, 1977.

[19] P. Paule. Greatest factorial factorization and symbolic summation. *J. Symb. Comput.*, 20(3):235–268, Sept. 1995.

[20] R. Pirastu. *On combinatorial identities: symbolic summation and umbral calculus*. PhD thesis, Johannes Kepler Universität, Linz, Austria, 1996.

[21] C. Tweedie. Nicole's contribution to the foundations of the calculus of finite differences. *Proceedings of the Edinburgh Mathematical Society*, 36:22–39, 2 1917.

Процесс обращения оператора взятия разности в классической литературе назывался “конечным интегрированием” или просто “интегрированием”:

“The operation of integration is therefore by definition the inverse of the operation denoted by the symbol Δ. As such it may with perfect propriety be denoted by the inverse form Δ^{-1}. It is usual however to employ for this purpose a distinct symbol, Σ, the origin of which, as well as of the term integration by which its office is denoted, it will be proper to explain.” — George Boole (1860) [7]

Поскольку здесь Буль не дает ссылки на раннее употребление знака суммирования, будет уместным упомянуть, что обозначение Σ для операции неопределенного сумирования было введено Эйлером:

“Just as we used the symbol Δ to signify a difference, so we use the symbol Σ to indicate a sum.” — Leonard Euler (1755) [9]

1st Form. Factorial expressions of the form $x(x-1)\dots(x-m+1) = x^{(m)}$. We have

$$\Delta x^{(m+1)} = (m+1)x^{(m)};$$

$$\text{therefore}\quad \Sigma x^{(m)} = \frac{x^{(m+1)}}{(m+1)} + C.$$

Thus also, if $u(x) = ax+b$, we have

$$\Sigma u(x)u(x-1)\dots u(x-m+1) = \frac{u(x)u(x-1)\dots u(x-m)}{(m+1)a} + C. \quad (3)$$

2nd Form. Rational and integral function of x. [a]

For, by Chap. II. Art. 5, any such function is reducible to a series of factorials of the preceding form, each of which may be integrated separately. We find for $\Sigma u(x)$ the general theorem

$$\Sigma u(x) = C + u(0)x + \Delta u(0)\frac{x^{(2)}}{1\cdot 2} + \Delta^2 u(0)\frac{x^{(3)}}{1\cdot 2\cdot 3} + \&c. \qquad (4)$$

which terminates when $u(x)$ is rational and integral. [b]

It is obvious that a rational and integral function of x may also be integrated by assuming for its integral a similar function of a degree higher by unity but with arbitrary coefficients whose values are to be determined by the condition that the difference of the assumed integral shall be equal to the function given.

---

[a] Термин *целая рациональная функция* употреблялся вместо термина *полином* в 18-м и 19-м веке.

[b] Здесь $\Delta^k u(0)$ – это k-я разность $u(x)$ в 0.

3rd Form. Factorial expressions of the form

$$\frac{1}{u(x)u(x+1)\ldots u(x+m)},$$

where $u(x)$ is of the form $ax+b$.

We have corresponding to the above form of $u(x)$

$$\Delta\frac{1}{u(x)u(x+1)\ldots u(x+m-1)}=\frac{-am}{u(x)u(x+1)\ldots u(x+m)}.$$

Hence $\Sigma\frac{1}{u(x)u(x+1)\ldots u(x+m)}=-\frac{1}{am(u(x)u(x+1)\ldots u(x+m-1))}.$

$$(5)$$

It will be observed that there must be at least two factors in the denominator of the expression to be integrated. No finite expression for $\Sigma\frac{1}{ax+b}$ exists.

To the above form certain more general forms are reducible. Thus we can integrate any rational fraction of the form

$$\frac{\phi(x)}{u(x)u(x+1)\dots u(x+m)},$$

$u(x)$ being of the form $ax+b$, and $\phi(x)$ a rational and integral function of x of a degree lower by at least two unities than the degree of the denominator. For, expressing $\phi(x)$ in the form

$$\phi(x) = Au(x)+Bu(x)u(x+1)+Cu(x)u(x+1)u(x+2)+\cdots+Eu(x)u(x+1)\cdots u(x+m-$$

A, B ... being constants to be determined by equating coefficients, or by an obvious extension of the theorem of Chap. II. Art. 5, we find

$$\Sigma\frac{\phi(x)}{u(x)u(x+1)\dots u(x+m)} = A\Sigma\frac{1}{u(x+1)u(x+2)\cdots u(x+m)}+$$
$$+B\Sigma\frac{1}{u(x+2)u(x+3)\cdots u(x+m)}+\cdots+E\Sigma\frac{1}{u(x+m-1)u(x+m)}, \tag{6}$$

and each term can now be integrated by (5).

As all that is known of the integration of rational functions is virtually continued in the two primary theorems of (3) and (5), it is desirable to express these in the simplest form. Supposing then $u(x) = ax + b$, let

$$u(x)u(x-1)\ldots u(x-m+1) = (ax+b)^{(m)},$$

$$\frac{1}{u(x)u(x+1)\ldots u(x+m-1)} = (ax+b)^{(-m)},$$

then

$$\Sigma(ax+b)^{(m)} = \frac{(ax+b)^{(m+1)}}{a(m+1)} + C, \tag{7}$$

whether m be positive or negative. [a] The analogy of this result with the theorem

$$\int (ax+b)^m = \frac{(ax+b)^{m+1}}{a(m+1)} + C,$$

is obvious.

[a] В предположении, что $m \neq -1$

4th Form. Functions of the form $a^x\phi(x)$ in which $\phi(x)$ is rational and integral. Since $\Delta a^x = (a-1)a^x$, we have

$$\Sigma a^x = \frac{a^x}{a-1} + C.$$

Applying integration by parts we have

$$\Sigma a^x \phi(x) = \frac{1}{a-1}\left\{\phi(x)a^x - a\Sigma a^x \Delta\phi(x)\right\}. \tag{8}$$

Thus the integration of $a^x\phi(x)$ is made to depend upon that of $a^x\Delta\phi(x)$; this again will by the same method depend upon that of $a^x\Delta^2\phi(x)$, and so on. Hence $\phi(x)$ being by hypothesis rational and integral, the process may be continued until the function under the sign Σ vanishes. This will happen after $n+1$ operations if $\phi(x)$ be of the nth degree; and the integral will be obtained in finite terms.[a]

[a]Здесь, по существу, представлен рекурсивный алгоритм суммирования квази-полиномов, в предположении $a \neq 1$.

Метод неопределенного суммирования полиномов обычно приписывается Стирлингу. Однако, согласно Твиди [21]: "While engaged in a study of the Methodus Differentialis of Jas. Stirling (1730) I have been struck by the fact that Nicole's papers on the same subject, printed in the Memoires de l'Academie Royale des Sciences (Paris), appear to form a fitting prelude to the work published by Stirling. The dates of Nicole's Papers are 1717, 1723, 1724, 1727, and it is almost certain that Stirling was well acquainted with their contents..." После описания четырех работ Николе, Твиди заключает: "From the foregoing it seems natural to infer that Nicole was the first to introduce the Inverse Factorial Series. His first Memoir bears the date 30th January 1717. In the Philosophical Transactions for the months of July, August, and September 1717 (No. 353), there was published a Memoir entitled De Seriebus Infinitis Tractatus. Pars Prima. Auctore Petro Remundo de Monmort, R.S.S. Una cum Appendice et Additamento per D. Brook Taylor, R.S. Sec.

From this Memoir it is clear that both Montmort and Brook Taylor had at the same time been busy with similar ideas."

Montmort in particular shows how to find $\Sigma\phi(x)/(x+n)...(x+(p-1)n)$ in the same way as Nicole, i.e., by writing $\phi(x)$ in the form $A_0 + A_1x + A_2x(x+n) + \ldots$. He also shows how to sum $\Sigma 1/(x+a)(x+b)(x+c)\ldots$, where $a, b, c\ldots$ are positive integers, by multiplying the numerator and denominator by the product $(x+a+1)(x+a+2)\ldots(x+b-1)(x+b+1)\ldots$, etc., and then proceeding as above. He also gives an expansion of $1/(x+a)$ in the form $A/x + B/x(x+1) +$ etc., commonly described as Stirling's Series.

Brook Taylor in his Appendix shows in a masterly manner how to deduce by his method of Increments the conclusions obtained by Montmort. In the summation of $\Sigma\phi(x)/(x+a)(x+b)\ldots$ he points out that the degree of $\phi(x)$ must be less by 2 than that of the denominator. He then represents the fraction $\phi(x)/(x+a)(x+b)\ldots$ as a sum of partial fractions

$$\frac{A}{x+a} + \frac{B}{x+b} + \ldots \tag{9}$$

where

$$A + B + \cdots \equiv 0. \tag{10}$$

35. We should observe this method carefully, since sums of differences of this kind cannot be found by the previous method. If the difference has a numerator or the denominator has factors that do not form an arithmetic progression, then the safest method for finding sums is to express the fraction as the sum of partial fractions. Although we may not be able to find the sum of an individual fraction, it may be possible to consider them in pairs. We have only to see whether it may be possible to use the formula

$$\Sigma\frac{1}{x+(n+1)\omega}-\Sigma\frac{1}{x+n\omega}=\frac{1}{x+n\omega}. \tag{11}$$

Although neither of these sums is known, still their difference is known.

36. In these cases the problem is reduced to finding the partial fractions, and this is treated at length in a previous book. [a]

[a] Here ω is an increment of independent variable x and can be replaced by 1.

Пример:

$$\Sigma\frac{3}{x(x+3)}=\Sigma\frac{1}{x}-\Sigma\frac{1}{x+3}=\Sigma\frac{1}{x+1}-\Sigma\frac{1}{x+3}-\frac{1}{x}=$$

$$=\Sigma\frac{1}{x+2}-\Sigma\frac{1}{x+3}-\frac{1}{x}-\frac{1}{x+1}=-\frac{1}{x}-\frac{1}{x+1}-\frac{1}{x+2}.$$

Эти идеи могут быть легко распространены на более сложные выражения:

$$\Sigma\frac{a^{x+1}}{x+1}-\Sigma\frac{a^x}{x}=\frac{a^x}{x},$$

или

$$\Sigma\frac{x+1}{x^3+3\,x^2+4\,x+3}-\Sigma\frac{x}{x^3+x+1}=\frac{x}{x^3+x+1}.$$

Пусть $F = f/g \in \mathbb{K}(x)$, с взаимно простыми $f, g \in \mathbb{K}[x] \setminus \{0\}$ и $\deg f < \deg g$.

Обозначим ρ *дисперсию* F – максимальное целое расстояние между корнями знаменателя g.

Если $\rho = 0$, то в (2) можно взять $R = 0$ и $H = F$ (Сергей Александрович Абрамов, 1971).

Пусть теперь $\rho > 0$.

Хорошо известно, что битовая длина суммы $G(x)$ в (1) может экспонециально зависеть от битовой длины сумманда $F(x)$.

$$F(x) = \frac{1}{x^2 + \rho x}$$

$$G(x) = -\frac{1}{\rho}\left(\frac{1}{x} + \frac{1}{x+1} + \cdots + \frac{1}{x+\rho-1}\right),$$

Имеется два вида зависимости времени выполнения алгоритмов суммирования от дисперсии в случаях, когда ρ имеет значение, экспоненциально зависящее от размера входа:

- *существенная* (неустраняемая) зависимость: алгоритм по меньшей мере линеен по ρ^ϵ для некотрого $0 < \epsilon \leq 1$ и ответ имеет битовую длину также линейную по ρ^ϵ;

- *несущественная* (потенциально устраняемая) зависимость: алгоритм по меньшей мере линеен по ρ^ϵ для некотрого $0 < \epsilon \leq 1$, но ответ имеет битовую длину полиномиально зависящую от размера входа.

Существенная зависимость является свойством конкретной задачи суммирования.
Несущественная зависимость является свойством конкретного алгоритма решения задачи суммирования.

$$\sum_x \frac{-2\,x+999}{(x+1)\,(x-999)\,x\,(x-1000)} = \frac{1}{x\,(x-1000)}, \tag{12}$$

$$\sum_x \frac{x^3-1998\,x^2+996999\,x+999999}{(x+1)\,(x-999)\,x\,(x-1000)} = \frac{1}{x\,(x-1000)}+\sum_x \frac{1}{x}. \tag{13}$$

Дисперсия в этих примерах равна 1001.

Абрамов

In his classical work [1] Sergei Abramov first introduced the notion of dispersion, and described an algorithm for rational indefinite summation. The algorithm was implemented in Lisp and used author's own polynomial arithmetic package. He later provided an algorithm to solve problem **P1** [2]. This result was somehow unknown for some period of time.

It is worthwhile to mention here the work of Moenck [18] in which he tries to generalize (6) to the case of denominators with factors of arbitrary degree. Moenck's algorithm was used in Maple for some period of time until some serious flaw was discovered in it.

Abramov's work becomes popular and widely adopted by computer algebra systems two decades later.

- **Iterative** (Hermite reduction analogous) algorithms will start with $R = 0$ and $H = F$ and decrease the dispersion of H by one at each iteration, reducing the non-rational part H and growing the rational part R. The number of iterations is equal to ρ, see [2].

- **Linear algebra based** (Ostrogradsky analogous) algorithms first build universal denominators u and v such that the denominator of R will divide u and the denominator of H will divide v. Then, the problem is reduced to solving a system of linear equations with size $\deg u + \deg v$, see [19, 3, 20]. Since $\deg u \geq \rho$ the choice of u of the lowest possible degree is obviously crucial here. The idea to build an universal denominator here is essentially the same as multiplying the numerator and denominator of the summand by missing factors in Boole's description above.

Госпер

Прямые методы:

Any given proper reduced rational function $f(x)/g(x)$ $(f(x), g(x) \in \mathbb{K}[x])$ can be uniquely represented as

$$\frac{f(x)}{g(x)} = \sum_{i=1}^{v} \sum_{j=1}^{m_i} \sum_{k=0}^{n_{ij}} \frac{f_{ijk}}{E^k g_i^j}, \tag{14}$$

where:

(i) g_i are monic distinct factors, irreducible over $\overline{\mathbb{K}}$, or
(ii) g_i are monic distinct factors, irreducible over $\mathbb{K}$, or
(iii) g_i are monic distinct factors, shiftless over $\mathbb{K}$.
Here v is the number of different shift equivalence classes (components) in the denominator of the summand; m_i is the highest multiplicity of a factor in class i; n_{ij} – the largest shift of a factor of multiplicity j in class i. All polynomials f_{ijk} have $\deg f_{ijk} < \deg g_i$. This means in particular that in case **(i)** $\deg f_{ijk} = 0$ and $f_{ijk} \in \overline{\mathbb{K}}$.

For the simplicity of description of the direct algorithm, (14) can be rewritten in the form

$$\frac{f(x)}{g(x)} = \sum_{i=1}^{v} \sum_{j=1}^{m_i} M_{ij}(E) \frac{1}{g_i^j}, \tag{15}$$

where $M_{ij}(E)$ is a linear difference operator with coefficients in $\overline{\mathbb{K}}$ or in $\mathbb{K}[x]$. This representation is unique in each of the cases **(i)** – **(iii)**. If $\frac{f(x)}{g(x)}$ is summable, the summable part $R(x)$ can be also uniquely written in a form analogous to (15):

$$\sum_{i=1}^{v} \sum_{j=1}^{m_i} L_{ij}(E) \frac{1}{g_i^j}, \tag{16}$$

and therefore

$$(E-1)L_{ij}(E) = M_{ij}(E), \tag{17}$$

i.e., every operator $M_{ij}(E)$ in (15) is left-divisible by $E-1$.

Let $M_{ij}(E) = a_p E^p + a_{p-1} E^{p-1} + \ldots + a_1 E + a_0$. Then the remainder from the left division of $M_{ij}(E)$ by $E - 1$ is simply

$$r_{ij} = E^{-p} a_p + E^{-(p-1)} a_{p-1} + \ldots + E^{-1} a_1 + a_0. \tag{18}$$

The summability criterion states that the polynomials r_{ij} in (18) must be identically equal to zero for all i, j in order for the input rational function to be summable.

Observe that

(i) requires factorization of the denominator into linear factors,

(ii) requires factorization in $\mathbb{K}[x]$ but does not require computation of the dispersion of the denominators, and

(iii) does not require factorization in $\mathbb{K}[x]$ but does require the knowledge of the so-called dispersion set (the set of all integer distances between the roots of the denominator).

The first description of direct rational summation algorithm is due to Karr [15] (see also popular survey of Lafon [16], where above-mentioned summability criterion is formulated explicitly with proper reference to Karr's work). Because – at that time – factorization was considered time-consuming it was effectively forgotten, and factorization-free (gcd-based) algorithms were used in computer algebra systems for years.

Decomposition **(i)** was used in [3] to establish a summability criterion and to describe the structure of a universal denominator. This criterion is the same as (10) since a linear difference operator with constant coefficients is left-divisible by $E - 1$ if and only if the sum of coefficients is equal to 0. This is also equivalent to the classical condition of summability which is that the degree of the numerator is at least 2 less than the degree of the denominator.

Representation **(iii)** with a fast algorithm to compute dispersion set was used in [12]. It provided the first polynomial time rational function summability test, and avoided any intermediate expression swell. If the output is exponentially large in the input size, the only part of the algorithm that exponentially depends on the input size is writing the result in expanded form. This was the first rational indefinite summation algorithm with only essential dependency of the running time on dispersion of the input for the case of summable input.

Example 1. Consider the following two summation problems:

$$\sum_x T(x) = \frac{5^x}{(x+1)(x+200)}, \tag{19}$$

where $T(x) = \dfrac{2(2x^2+401x+299)5^x}{(x+1)(x+2)(x+200)(x+201)}$ and

$$\sum_x \frac{\left(9\,x^4+1434\,x^3+70075\,x^2+1017440\,x-252800\right)5^x}{(x+40)\,(x+80)\,(x+79)\,(x+1)\,x} = \tag{20}$$

$$\frac{5^x\,(2\,x+79)}{(x+79)\,x} + \sum_x \frac{5^x}{x+40}$$

For the sum (19) the Gosper-Petkovšek form of the certificate of the summand consists of polynomials $Q(x) = 5(x+1)$,
$R(x) = x + 202$ and
$P(x) = \left(x^2 + \frac{401}{2}x + \frac{299}{2}\right)(x+199)(x+198)\cdots(x+3)$.
This last polynomial has degree one less than the dispersion of the input (199 in this case), and is saturated. After finding a polynomial solution $U(x)$ (with degree bound 199) of the Gosper equation

$$Q(x)U(x+1) - R(x-1)U(x) = P(x),$$

one forms the final result as

$$\frac{Q(x-1)U(x)}{P(x)}T(x) = \frac{5^x}{(x+1)(x+200)},$$

and most of the terms of $P(x)$ and $U(x)$ cancel.

For the sum (20) Gosper's algorithm will not return any answer (since the input is non-summable), so the general additive decomposition method from [5] is used. It is worthwhile to mention that this algorithm uses a number of steps at least linear in the dispersion of the input (which is 80 in this particular case), and returns the result, whose summable part has the numerator of degree 40, the denominator of degree 41, and the non-summable part is

$$\sum_x 9094947017729282379150390625 \, \frac{5^x}{x+80}.$$

Unlike traditional algorithms, our direct algorithms are based on a change of representation of the summand.

0.1 Polynomials and quasi-polynomials

In the calculus of finite differences it is always advantageous to express polynomials by Newton's formula. – Charles Jordan (1939)

In case of indefinite summation of (quasi-)polynomials special representation means just change of basis. The change from monomial basis to the falling factorial basis is a common place in many sources to provide algorithms for polynomial summation, that involves either computation of Stirling numbers, or Bernoulli polynomials (see for example [16, 10]). However, the binomial basis $\binom{x}{i}, i = 0, 1, \ldots, n$ happens to be the most suitable in this context. Indeed, assume all polynomials are given in the binomial basis, i.e., a polynomial $P_n(x)$ of degree n is represented by a list $\varphi_0, \varphi_1, \ldots, \varphi_n \in \mathbb{K}$ such that

$$[\varphi_0, \varphi_1, \ldots, \varphi_n](x) = \varphi_0 \binom{x}{0} + \varphi_1 \binom{x}{1} + \ldots + \varphi_n \binom{x}{n} \qquad (21)$$

Addition, subtraction and multiplication by a constant in binomial basis are componentwise (similarly to the monomial basis). Note that $\varphi_0, \varphi_1, \ldots, \varphi_n$ in (21) are just finite forward differences (elements of the Newton series, or pure-plus chains of recurrences) of $P_n(x)$ taken at $x = 0$ with step 1.Recall

$$E[\varphi_0, \varphi_1, \ldots, \varphi_n] = [\varphi_0 + \varphi_1, \varphi_1 + \varphi_2, \ldots, \varphi_{n-1} + \varphi_n, \varphi_n]$$

$$(E - 1)[\varphi_0, \varphi_1, \ldots, \varphi_n] = [\varphi_1, \varphi_2, \ldots, \varphi_n]$$

$$\sum_x [\varphi_1, \varphi_2 \ldots, \varphi_n] = [\varphi_0, \varphi_1, \varphi_2, \ldots, \varphi_n]$$

for an arbitrary constant φ_0.

This suggests a constant time indefinite summation algorithm in binomial basis, which can be expressed in Maple (assuming that input polynomial is represented by the list of coefficients in binomial basis and summation variable is implicit) as:

```
poly_sum := y->[0,op(y)];
```

This is probably the shortest and fastest algorithm for indefinite summation, and it is not surprising that it was known long before the first computer algebra system was even prototyped (see the example on page 103 of [14], which is the third edition of the original book published in 1939). However we are not aware of any implementation of this algorithm in a computer algebra system. Although, even for manual manipulation of the problem the formula $\sum_x \binom{x}{m} = \binom{x}{m+1} + C$ is better than the popular $\sum_x x^{(m)} = \frac{x^{(m+1)}}{(m+1)} + C$.

Just compare the right-hand sides to see how to save 6 symbols in the output. Note that in more general context the binomial basis was used in [4] where it is shown why the binomial basis is the basis of choice when solving linear difference equations with polynomial coefficients. However Maple standard summation routines ignore this fact.

Representation of polynomials in binomial basis is especially appealing, because it can be used as a standard dense representation for polynomials. A lengthy chain of computations involving ring operations, (nested) indefinite summation, differencing, etc. can be performed in this basis from the beginning. If output in standard basis is required, conversion may be performed when necessary. On one hand, all ring operations have the same asymptotic time complexity in both bases (see [11]). On the other hand, operations of differencing and indefinite summation have time complexity bounded by a constant (the best one can hope for), and shift is performed in linear time.

We strongly agree with the old comment of Jordan on importance of Newton's formula for the statistician:

This is not yet sufficiently recognized, since nearly always the statistician expands his polynomial in power series in spite of the fact that he is generally concerned with the differences and sums of

his functions, so that he is obliged to calculate these quantities laboriously. In Newton's expansion they would be given immediately. (see [13])
This is especially true for Maple, because, for example, Newton interpolation routines return the resulting interpolating polynomial effectively in Newton form.

Consider now a quasi-polynomial with polynomial part represented in binomial basis:

$$q(x) = \lambda^x[\varphi_0, \varphi_1, \ldots, \varphi_n],$$

where $\lambda \neq 1$, $\lambda \neq 0$. It is well known (see for example [17]) that the indefinite sum of $q(x)$ has polynomial part of the same degree n, i.e.:

$$\sum_x q(x) = r(x) = \lambda^x[\xi_0, \xi_1, \ldots, \xi_n]. \tag{22}$$

Now,

$$Er(x) = \lambda^{x+1}[\xi_0 + \xi_1, \xi_1 + \xi_2, \ldots, \xi_{n-1} + \xi_n, \xi_n] =$$

$$\lambda^x[\lambda\xi_0 + \lambda\xi_1, \ldots, \lambda\xi_{n-1} + \lambda\xi_n, \lambda\xi_n],$$

$$(E-1)r(x) = \lambda^x[(\lambda-1)\xi_0 + \lambda\xi_1, \ldots, (\lambda-1)\xi_{n-1} + \lambda\xi_n, (\lambda-1)\xi_n],$$

and application of the operator $(E-1)$ to l.h.s. and r.h.s. of (22) gives

$$[\varphi_0, \varphi_1, \ldots, \varphi_n] = [(\lambda - 1)\xi_0 + \lambda\xi_1, \ldots, (\lambda - 1)\xi_{n-1} + \lambda\xi_n, (\lambda - 1)\xi_n],$$

i.e., the coefficients of the polynomial part of the indefinite sum can be computed as

$$\xi_n = \frac{\varphi_n}{\lambda - 1}, \xi_i = \frac{\varphi_i - \lambda\xi_{i+1}}{\lambda - 1}, i = n - 1, n - 2, \ldots, 0.$$

This gives a linear time algorithm for indefinite summation of quasi-polynomials.

If the input is given in the monomial basis, then the running time of summation is again dominated by the time to convert the input to binomial basis and back. The current implementation in Maple is at least quadratic in the degree of polynomial part of the summand.

0.2 Rational and quasi-rational functions

The safest method for finding sums is to express the fraction as the sum of partial fractions. Although we may not be able to find the sum of an individual fraction, it may be possible to consider them in pairs. — Leonard Euler (1742)

As a piece of computer algebra folklore we mention here that, to our knowledge, one of the most popular computer algebra systems has never used factorization-free algorithms for the rational summation. It always proceeded with partial fraction decomposition and used variations of Euler's simplification (11).

Contrary to this, Maple system for a long time had an obsession with factorization-free implementations of rational indefinite summation. In some versions of Maple this obsession has led to anecdotal situations. For example, Maple summation routine would compute the dispersion set of the denominator of the input summand using factorization, and then "forget" about the fact that the denominator was already factored to run factorization-free implementation of Abramov's algorithm, that requires dispersion set as the input.

0.3 Experiments

Polynomials

We are unaware of any implementation of fast basis conversion in Maple, and also were unable to obtain NTL implementation from authors of [8]. However Figure 4 in [8] gives a good idea of possible speedup of indefinite summation of polynomials. It compares timings of fast basis conversion with the naive basis conversion (which is essentially the time for basis conversion in our old implementation).

In order to show the potential gain from working with polynomials in the binomial basis, we compare timings in Maple to evaluate the following expression

$$V(x) = \sum_x \left(\sum_x P_1 + \left(\sum_x P_2 + P_1\right)\right)$$

with random polynomials P_1, P_2 [a]. The timing comparison is presented in Table 1. Note that for basis conversion we use our ancient implementation [6] of chains of recurrences which is quadratic in the degree of the polynomials.

n	`Maple`	`Prototype`	`incl. Basis_conv`
100	0.031	0.016	$<$0.016
200	0.140	0.047	0.031
400	0.578	0.312	0.272
800	4.617	1.794	1.698
1600	33.088	12.558	11.912

Таблица 1: Timings in seconds for $V(x)$

[a] All random polynomials in our experiments have integer coefficients between -99 and 99.

According to the results in [8], the last column (in this range of degrees) can be reduced by a factor between 2 and 6 if fast basis conversion is used.

Quasi-polynomials

The comparison (especially in the case of quasi-polynomials, and quasi-polynomials whose coefficients depend on λ) is very favorable for the direct algorithms. The following tables compare timings for summation of quasi-polynomials with random polynomial part performed by Maple 16 using `SumTools[Indefinite]`, `SumTools[Hypergeometric]` and our prototype. For the first and the second tables random dense polynomials $P_n(x)$ of degree n were generated, for the third table random dense polynomials in $P_n(x, \lambda)$ of degree n were used. A dash in the table indicates that Maple 16 did not return a result after 1000 seconds, and "Err" indicates that Maple 16 returned an error. Most of the time reported by our prototype was spent in basis conversion. As soon as fast basis conversion is implemented in Maple, these timings will improve tremendously (see Figure 4 in [8]).

n	`Indefinite`	`Hypergeometric`	`Quasi_Poly`
20	0.031	0.078	$<$0.016
40	0.156	0.078	$<$0.016
80	1.264	0.234	0.016
160	14.805	1.202	0.016
320	40.888	4.883	0.156

Таблица 2: Timings in seconds for $5^x P_n(x)$

n	`Indefinite`	`Hypergeometric`	`Quasi_Poly`
20	0.078	0.484	<0.016
40	0.796	7.893	<0.016
80	9.516	442.965	0.016
160	81.105	–	0.031
320	Err	–	0.218

Таблица 3: Timings in seconds for $\lambda^x P_n(x)$

n	`Indefinite`	`Hypergeometric`	`Quasi_Poly`
20	0.109	0.687	<0.016
40	0.967	7.941	0.031
80	12.246	415.322	0.124
160	101.478	–	1.139
320	290.208	–	10.156

Таблица 4: Timings in seconds $\lambda^x P_n(x, \lambda)$

Quasi-rational functions

Example (19) takes 0.031 seconds with our implementation, 0.889 with
`SumTools[IndefiniteSum][Indefinite]` in Maple 16, and 0.265 seconds with
`SumTools[Hypergeometric][SumDecomposition]`. Example (20) takes 0.031 seconds with our implementation and provides minimal degree of the denominator of the summable part, 0.437 with
`SumTools[IndefiniteSum][Indefinite]` in Maple 16, and 0.390 seconds with
`SumTools[Hypergeometric][SumDecomposition]` and returns summable part with the denominator of degree 41 with standard Maple implementation.

Note that these examples have relatively small value of the dispersion. The speedup can be made arbitrarily large, by increasing the value of the dispersion. For instance, the sum

$$\sum_x \frac{\left(8\,x^3 + 12006\,x^2 + 4005998\,x - 1001000\right) 5^x}{x^4 + 2002\,x^3 + 1003001\,x^2 + 1001000\,x}$$

is evaluated to

$$2\,\frac{5^x\,(x+500)}{x\,(x+1000)}$$

in 0.047 seconds by our code, and in 6.849 seconds by `SumTools[Hypergeometric][SumDecomposition]`.

О роли дисперсии