

FPS In Action: An Easy Way To Find Explicit Formulas For Interlaced Hypergeometric Sequences

Bertrand Teguia Tabuguia
 Nonlinear Algebra Group
 Max Planck Institute for Mathematics in the Sciences
 04103 Leipzig, Germany
 bertrand.teguia@mis.mpg.de

Wolfram Koepf
 Mathematics and Natural Sciences
 University of Kassel
 34132 Kassel, Germany
 koepf@mathematik.uni-kassel.de

Abstract

Linear recurrence equations with constant coefficients define the power series coefficients of rational functions. However, one usually prefers to have an explicit formula for the sequence of coefficients, provided that such a formula is “simple” enough. Simplicity is related to the compactness of the formula due to the presence of algebraic numbers: “the smaller, the simpler”. This poster showcases the capacity of recent updates on the Formal Power Series (FPS) algorithm, implemented in Maxima and Maple (`convert/FormalPowerSeries`), to find simple formulas for sequences like those from <https://oeis.org/A307717>, <https://oeis.org/A226782>, or <https://oeis.org/A226784> by computing power series representations of their correctly guessed generating functions. We designed the algorithm for the more general context of univariate P -recursive sequences. Our implementations are available at http://www.mathematik.uni-kassel.de/~bteguia/FPS_webpage/FPS.htm.

Keywords Hypergeometric type power series, D -finite function, m -fold hypergeometric term, P -recursive sequence, Guessing, Explicit formula

1 Introduction

Let \mathbb{K} be a field of characteristic zero, usually a finite extension field of the rationals. We call power series representation for a function $f(x) = \sum_{n=0}^{\infty} a_n x^n$, $a_n \in \mathbb{K}$, a formula for the $(n+1)^{\text{st}}$ summand $a_n x^n$ that one can use to compute any truncation of the power series of f . The latter coincide with the Taylor expansion of f when f is analytic at the origin, and thus the representation could also be defined at any point $x_0 \in \mathbb{K}$ where f is analytic. In the univariate case, a function f is called holonomic or D -finite, if it satisfies a linear differential equation with polynomial coefficients (holonomic DE). From [6], we have a general strategy to search for power series representations symbolically. Given a holonomic function $f(x) = \sum_{n=0}^{\infty} a_n x^n$, as an expression in the variable x ,

1. compute a holonomic DE satisfied by $f(x)$;
2. convert the holonomic DE into a linear recurrence equation with polynomial coefficients (holonomic RE);
3. *solve* the holonomic RE for the coefficients a_n .

The two last steps are equivalent to finding power series solutions of linear ordinary differential equations. One cannot always solve the resulting recurrence equation.

For $m \in \mathbb{N}$, a term a_n is called m -fold hypergeometric if the ratio $a_{n+m}/a_n \in \mathbb{K}(n)$. When m is not specified, m -fold hypergeometric denotes all such terms for arbitrary positive integers m . We say that a

function is of hypergeometric type if its power series coefficients are evaluations of m -fold hypergeometric terms. The definition extends to Laurent-Puiseux series, but in this paper we only consider power series with non-negative integer exponents. Numerous holonomic functions, including rational functions, are of hypergeometric type. The computation of m -fold hypergeometric term solutions of holonomic REs is effective (see [9, 11, 14]). On this poster, we use the algorithm `mfoldHyper` (available in Maple 2022 as `LREtools-mhypergeomsols`), developed in [14] for the efficient computation of hypergeometric type power series (see also [13, 12]). It is worth mentioning that `mfoldHyper` extends the algorithms by Petkovšek and Mark van Hoeij (see [7, 15, 1]) and has a much better performance than some previous approaches in the same direction (see [8, 3]). Our Formal Power Series (FPS) algorithm uses `mfoldHyper` to compute a basis for all m -fold hypergeometric term solutions on the third step given above, and uses linear algebra with a truncated series expansion of f to deduce a power series representation for f . For more details about the algorithm, we refer the reader to [14, 11].

Example 1. *The FPS algorithm computes the following power series representation for $\frac{1}{((x^2-p)(x^3-q))}$ for arbitrary constants p and q .*

$$\begin{aligned} & \sum_{n=0}^{\infty} \left(-\frac{\left(qp^{-1-\frac{n}{2}} - q^{-\frac{1}{3}-\frac{n}{3}}p \right) x^n}{p^3 - q^2} \right) + \sum_{n=0}^{\infty} \left(-\frac{\left(p^{\frac{3}{2}} - q \right) p^{-n-\frac{3}{2}} x^{2n+1}}{p^3 - q^2} \right) \\ & + \sum_{n=0}^{\infty} \left(-\frac{q^{-1-n} p \left(q^{\frac{2}{3}} - p \right) x^{3n}}{p^3 - q^2} \right) + \left(\sum_{n=0}^{\infty} \frac{\left(q^{\frac{2}{3}} - p \right) q^{-n-\frac{2}{3}} x^{3n+1}}{p^3 - q^2} \right). \end{aligned} \quad (1)$$

For software reason, the formula is not identical with the one obtained with Maple 2022, although correct; however, the output is still much more compact compared to previous Maple versions. Our implementation is available for Maple and Maxima users at http://www.mathematik.uni-kassel.de/~bteguia/FPS_webpage/FPS.htm. The authors welcome any comments for the improvement of the package.

The latter example presents a typical situation of what happens when it comes to computing explicit formulas for power series coefficients of rational functions that generate some sequences from N. J. A. Sloane <https://oeis.org>. FPS splits the formula modulo some integers, which allows to also deal with the situation of many zeros (see [5]) in the sequence with no specific care about them. The Padé approximation is often the best choice to guess the rational function that generates a sequence [2]. In our next examples, we use the implementation in the `Gfun` package (`ratpoly`) ([10]) to find the generating functions, and FPS to compute the desired formulas.

2 Some Explicit Formulas

In what follows, we compute power series representations of rational functions generating some sequences from <https://oeis.org>. The resulting representations give explicit formulas for the $(n+1)^{\text{st}}$ terms (we start sequences at 0) of the corresponding sequences. We do not give detailed proofs that the guessed rational function is the correct one; however, the formula obtained using FPS is correct by the correctness of the FPS algorithm. Throughout this section a_n will denote the $(n+1)^{\text{st}}$ term of the sequence for each example.

Example 2. *Let us consider the sequence **A307717** from <https://oeis.org/A307717>. a_n counts the number of palindromic squares, k^2 , of length $n+1$ such that k is also palindromic.*

We use the 33 first terms of the sequence. That is the minimal required by the guess in the next step.

> L:=[4, 0, 2, 0, 5, 0, 3, 0, 8, 0, 5, 0, 13, 0, 9, 0, 22, 0,
16, 0, 37, 0, 27, 0, 60, 0, 43, 0, 93, 0, 65, 0, 138]:

We guess the generating function.

> f:=gfun:-listtoratpoly(L,x)[1]

$$f := -\frac{2x^{16} - x^{14} - 5x^{12} + 5x^{10} + 12x^8 - 5x^6 - 11x^4 + 2x^2 + 4}{-x^{16} + 4x^{12} - 6x^8 + 4x^4 - 1} \quad (2)$$

We compute the power series representation using FPS. The input is the expression f , its variable x , and a summation variable n chosen by the user.

> FPS(f,x,n)

$$2 + \left(\sum_{n=0}^{\infty} \left(\frac{(-1)^n n^3}{96} - \frac{3(-1)^n n^2}{32} + \frac{65(-1)^n n}{96} + \frac{7(-1)^n}{32} + \frac{n^3}{32} - \frac{5n^2}{32} + \frac{37n}{32} + \frac{57}{32} \right) x^{2n} \right) \quad (3)$$

Theorem 3. The sequence **A307717** from <https://oeis.org/A307717> has the explicit formula

$$a_0 = 4$$

$$a_{2n+1} = 0, n \geq 0 \quad (4)$$

$$a_{2n} = \frac{((-1)^n + 3)n^3 - (9(-1)^n + 15)n^2 + (65(-1)^n + 111)n + 21(-1)^n + 171}{96}, n \geq 1, \quad (5)$$

where a_n is its $(n+1)^{st}$ term.

Proof. It is enough to prove that the generating function (2) is the correct one. From the connection to the sequence from <https://oeis.org/A218035> whose generating function, denote it by $g(x)$, is known, one verifies that f in (2) and g are linked by the relation $f(x) = g(x^2)/x^2$, which holds. \square

We noticed that the latter example is also investigated in the recent paper [4] about guessing. There the authors also provide a different formula for **A307717**.

Example 4. Our second sequence is **A226782** from <https://oeis.org/A226782>. $a_n = 0$ if $n+1$ is even, and the inverse of 4 in the ring $\mathbb{Z}/(n+1)\mathbb{Z}^*$ if $n+1$ is odd. We proceed as before and find an explicit formula for a_n .

> L:=[0, 0, 1, 0, 4, 0, 2, 0, 7, 0, 3, 0, 10, 0, 4, 0, 13]:

> f:=gfun:-listtoratpoly(L,x)[1]

$$f := -\frac{-x^8 + 4x^4 + x^2}{-x^8 + 2x^4 - 1} \quad (6)$$

> FPS(f,x,n)

$$-1 + \left(\sum_{n=0}^{\infty} \left(\frac{(-1)^n n}{2} + \frac{(-1)^n}{4} + n + \frac{3}{4} \right) x^{2n} \right) \quad (7)$$

In this example, the representation can be rewritten without the extra -1 since the series part evaluates to 1 at $n = 0$.

Theorem 5. *The sequence A226782 from <https://oeis.org/A226782> has the explicit formula*

$$\begin{aligned} a_0 &= 0 \\ a_{2n+1} &= 0, n \geq 0 \end{aligned} \tag{8}$$

$$a_{2n} = \frac{(-1)^n n}{2} + \frac{(-1)^n}{4} + n + \frac{3}{4}, n \geq 1, \tag{9}$$

where a_n is its $(n+1)^{st}$ term.

Proof. One verifies that the generating function given in <https://oeis.org/A226782> is a shift of f in (6), as we start the sequence at 0 instead of 1. \square

3 Conclusion

This poster aims to recommend our FPS implementation to the community of computer algebraists or scientists interested in finding explicit formulas for sequences. Combining it with guessing strategies enables one to discover or recover explicit formulas. Of course, limitations may always exist, at least in practice, when implementing the theory; however, as research goes on, FPS also keeps improving to produce explicit formulas for larger classes of functions in the future. We mention that previous versions of the `convert/FormalPowerSeries` package could not compute the representations presented here. However, this is now possible with Maple 2022, which incorporates our implementation thanks to Jürgen Gerhard from Maplesoft.

References

- [1] Thomas Cluzeau and Mark van Hoeij. 2006. Computing hypergeometric solutions of linear recurrence equations. *Appl. Algebra Engrg. Comm. Comput.* 17, 2 (2006), 83–115.
- [2] Keith O. Geddes. 1979. Symbolic computation of Padé approximants. *ACM Transactions on Mathematical Software (TOMS)* 5, 2 (1979), 218–233.
- [3] Peter Horn, Wolfram Koepf, and Torsten Sprenger. 2012. m-Fold hypergeometric solutions of linear recurrence equations revisited. *Mathematics in Computer Science* 6, 1 (2012), 61–77.
- [4] Manuel Kauers and Christoph Koutschan. 2022. Guessing with Little Data. *arXiv preprint arXiv:2202.07966* (2022).
- [5] Manuel Kauers and Thibaut Verron. 2019. Why you should remove zeros from data before guessing. *ACM Communications in Computer Algebra* 53, 3 (2019), 126–129.
- [6] Wolfram Koepf. 1992. Power series in computer algebra. *J. Symbolic Comput.* 13, 6 (1992), 581–603.
- [7] Marko Petkovšek. 1992. Hypergeometric solutions of linear recurrences with polynomial coefficients. *J. Symbolic Comput.* 14, 2-3 (1992), 243–264.
- [8] Marko Petkovšek and Bruno Salvy. 1993. Finding all hypergeometric solutions of linear differential equations. In *Proceedings of the 1993 international symposium on Symbolic and algebraic computation*. 27–33.
- [9] Anna A Ryabenko. 2002. Formal solutions of linear ordinary differential equations containing m -hypergeometric series. *Programming and Computer Software* 28, 2 (2002), 92–101.

- [10] Bruno Salvy and Paul Zimmermann. 1994. GFUN: a Maple package for the manipulation of generating and Holonomic functions in one variable. *ACM Trans. Math. Software* 20, 2 (1994), 163–177.
- [11] Bertrand Tegua Tabugua. 2020. *Power Series Representations of Hypergeometric Types and Non-Holonomic Functions in Computer Algebra*. Ph. D. Dissertation. University of Kassel, <https://kobra.uni-kassel.de/handle/123456789/11598>.
- [12] Bertrand Tegua Tabugua and W Koepf. 2021a. Hypergeometric type power series. In *4th International Conference “Computer Algebra”*. Dorodnicyn Computing Center, Federal Research Center “Computer Science and Control” of Russian Academy of Sciences (CCAS) and Peoples Friendship University of Russia, 105–108.
- [13] Bertrand Tegua Tabugua and Wolfram Koepf. 2021b. Power series representations of hypergeometric type functions. In *Maple in Mathematics Education and Research. MC 2020*. Editors: Corless R., Gerhard J., Kotsireas I., Communications in Computer and Information Science, Springer, 376–393.
- [14] Bertrand Tegua Tabugua and Wolfram Koepf. 2022. Symbolic conversion of holonomic functions to hypergeometric type power series. *Programming and Computer Software* 48, 2 (2022), 125–146.
- [15] Mark Van Hoeij. 1999. Finite singularities and hypergeometric solutions of linear recurrence equations. *J. Pure Appl. Algebra* 139, 1-3 (1999), 109–131.