Guessing With Quadratic Differential Equations

Bertrand Teguia Tabuguia

Nonlinear Algebra Group

Max Planck Institute for Mathematics in the Sciences

04103 Leipzig, Germany

`bertrand.teguia@mis.mpg.de`

Abstract

By holonomic guessing, we denote the process of finding a linear differential equation with polynomial coefficients satisfied by the generating function of a sequence, for which only a few first terms are known. Holonomic guessing has been used in computer algebra for over three decades to demonstrate the value of the guess-and-prove paradigm in intuition processes preceding proofs, as propagated in The Art of Solving (Polya, 1978). Among the prominent packages used to perform guessing, one can cite the Maple Gfun package of Salvy and Zimmermann; the Mathematica GeneratingFunctions package of Mallinger; and the Sage ore_algebra package of Kauers, Jaroschek, and Johansson.

We propose an approach that extends holonomic guessing by allowing the targeted differential equations to be of degree at most two. Consequently, it enables us to capture more generating functions than just holonomic functions. The corresponding recurrence equations are similar to known equations for the Bernoulli, Euler, and Bell numbers. As a result, our software finds the correct recurrence and differential equations for the generating functions of the up/down numbers (`https://oeis.org/A000111`), the evaluations of the zeta function at positive even integers, the Taylor coefficients of the Lambert W function, and many more. Our Maple implementation (`delta2guess`) is part of the FPS package which can be downloaded at `http://www.mathematik.uni-kassel.de/~bteguia/FPS_webpage/FPS.htm`.

Keywords Non-holonomic function, Recurrence equation, Cauchy product formula, Bernoulli numbers, Euler numbers, Bell numbers.

1   Introduction

Roughly speaking, guessing is finding the solution to a given problem by over-constraining the shape of a hypothetical solution from finite data. By its nice closure properties, univariate D-finite functions, or simply holonomic functions, stayed at the center of this method (see [5, 2, 6, 14, 10]). However, there are sequences and functions for which the holonomic system cannot help (see [1]). For instance, the reciprocal of a holonomic function is generally not holonomic (see [3]). Therefore sequences whose generating functions are reciprocals of holonomic functions are out of reach with the commonly used guessing. In [12, 13], the class of δ_2-finite functions was introduced. We recall the definition below.

Definition 1 (δ_2-finite functions). *Assume $\frac{d^{-1}}{dz^{-1}}f = 1$ for any function $f := f(z)$. Let $\delta_{2,z}$ be the operator defined as:*

$$\delta_{2,z}^k(f) = \frac{d^{i-2}}{dz^{i-2}}f \cdot \frac{d^{j-2}}{dz^{j-2}}f = f(z)^{(i-2)} \cdot f(z)^{(j-2)}, \quad where \;\; (i,j) = \nu(k). \tag{1}$$

$$\nu(k) = (i,j) = \begin{cases} (l,l) & if \quad N = k \\ (l+1, k-N) & otherwise \end{cases}, \tag{2}$$

$$\text{where } l = \left\lfloor \sqrt{2k + \frac{1}{4}} - \frac{1}{2} \right\rfloor, \quad \text{and} \quad N = \frac{l(l+1)}{2}. \tag{3}$$

A function $f(z)$ is called δ_2-finite if there exist polynomials $P_0(z), \cdots, P_d(z)$ over a field $\mathbb{K}$ of characteristic zero, not all zero, such that

$$P_d(z)\delta_{2,z}^{d+2}(f(z)) + \cdots + P_2(z)\delta_{2,z}^4(f(z)) + P_1(z)\delta_{2,z}^3(f(z)) + P_0(z)f(z) = 0. \tag{4}$$

The derivative operator $\delta_{2,z}$ in Definition 1 computes quadratic (degree 2) terms according to the lexicographic ordering in the differential polynomial ring $\mathcal{K}_f := \mathbb{K}[z][f(z), f(z)', f(z)'', f(z)^{(3)}, \ldots]$. In [11], finiteness relates to the finite-dimensional vector space that spans the derivatives. In our case, this relates to the finitely many variables of $\mathcal{K}_f$ for a given δ_2-finite function f. The author is currently working on more elaborations in this regard, including particular closure properties. The class of δ_2-finite functions contains holonomic functions. It also contains reciprocals of those satisfying second-order holonomic differential equations (see [13]). Functions satisfying first-order holonomic differential equations are holonomic by the arguments in [3], hence also δ_2-finite. Some interesting δ_2-finite functions are the Lambert W function, the reciprocals of the Bessel functions, and many generating functions involving numbers like the Bell, Bernoulli, and Euler numbers. The algorithm in [13] outputs power series representations in the form

$$f(z) = \sum_{n=0}^{\infty} a_n z^n, \begin{cases} a_{n+p} = \Phi(n, a_0, .., a_{n+p-1}), \\ \\ a_0, a_1.., a_{p-1} \text{ taken from the series truncation of order } p \end{cases}, \tag{5}$$

where $\Phi : \mathbb{Z} \times \mathbb{K}^{n+p} \longrightarrow \mathbb{K}$ is a recursive relation deduced from the recurrence equation associated to a quadratic (or δ_2-finite since it can be holonomic) differential equation of the form (4) satisfied by $f(z)$.

It is customary to find many applications of the study of sequences in enumerative combinatorics. These Bernoulli, Euler, and Bell numbers often occur in combinatorial problems. Thus, one thinks of an algorithm that will take a few terms $a_0, a_1, \ldots, a_N$ of a given sequence whose explicit formula might be unknown and find the corresponding Φ and its associated differential equation.

Previously, Hebisch and Rubey (see [4]) investigated guessing with the whole family of algebraic differential equations, which of course, includes the δ_2-finite one. However, we are not aware of a theoretical reason stipulating why one should investigate more arbitrary-degree differential polynomials than second-degree ones. We think that a systematic study of the relation between degree-k and degree-(k+1) differential polynomials will provide more insight into the use of algebraic differential equations. A simple reason to work on δ_2-finite equations is that most of the algebraic differential equations in [4] have their δ_2-finite analog – tempting to illustrate that δ_2-finite differential equations might suffice in many cases. Note, moreover, that lower-degree algebraic differential equations are usually desired when it comes to solving them using algebraic geometry techniques (see [8],[7, Chapter 3],[9]) or writing recursive formulas for power series solutions.

2   Sketch of the algorithm

We are given $N + 1$ terms $a_0, \ldots, a_N$ of an "unknown" sequence $(a_n)_{n \geqslant 0}$ over a field $\mathbb{K}$ of characteristic zero. We want to find a δ_2-finite differential equation

$$\sum_{k=0}^{d} P_k(z)\delta_{2,z}^{k+2}(f(z)) = 0, \tag{6}$$

such that $f(z) = \sum_{n=0}^{\infty} a_n z^n$. The algorithm proceeds as follows:

1. Fix a degree bound $m \in \mathbb{N}$ for the polynomial coefficients $P_k(z)$, $0 \le k \le d$. We use the default value $m = 2$ since this suffices to correctly guess the equations for the Bernoulli, Euler, and Bell numbers.

2. For $3 \le d \le \lceil (N+1)/(m+1) \rceil$

   (a) Compute the recurrence relation of the power series solutions of the ansatz

   $$\sum_{k=0}^{d} (c_{k,0} + \cdots + c_{k,m} z^m)\, \delta_{2,z}^{k+2}(f(z)), \tag{7}$$

   where $c_{k,i}, k = 0 \ldots d, i = 0 \ldots m$ are unknown coefficients.

   (b) Evaluate the recurrence equation for $n = 0, \ldots, (m+1)(d+1) - 1$, assuming $a_n = 0$ for $n \le -1$, and using the given first terms. This yields a linear system for the unknowns $c_{k,i}$'s that we solve using Maple linear system solver (`SolveTools:-Linear`) which implements many efficient algorithms for linear systems solving. Denote by S the set of solutions.

   (c) If S is not empty and $(m+1)(d+1) < N+1$, then use the $N+1-(m+1)(d+1)$ remaining initial terms to verify the solution found. Set S to the empty set if the verification fails. Interrupt the loop (of step 2 as a whole) if the verification succeeds.

3. Return the differential equation and the recurrence equation corresponding to the solution in S if $S \ne \emptyset$; otherwise return FAIL (not successful).

3 Examples

We implemented our algorithm in Maple with the name `delta2guess` (or `FPS:-delta2guess`) as a procedure of the `FPS` package available at `http://www.mathematik.uni-kassel.de/~bteguia/FPS_webpage/FPS.htm`. The procedure takes a list of finitely many numbers as input and returns a list of two elements in the successful case. These are a generic differential equation for the generating function and a generic recurrence equation for the input numbers, understood as the corresponding power series coefficients.

Example 2. *Maple does not, by default, compute $\zeta(50)$, where ζ denotes the Riemann Zeta function. Therefore this example is more interesting to test our implementation. Using our guessed recurrence equation, we can compute more terms of the sequence by no means of the Zeta function. The verification of the correctness is easier since $(\zeta(2n))_{n \geqslant 1}$ has a well-known explicit formula in terms of the Bernoulli numbers.*

$$> \ L := [\, seq(Zeta(2*j), j = 1..24)\,]: \ L[1..5]$$

$$\left[\frac{\pi^2}{6}, \frac{\pi^4}{90}, \frac{\pi^6}{945}, \frac{\pi^8}{9450}, \frac{\pi^{10}}{93555} \right] \tag{8}$$

$$> \ FPS:-delta2guess(L)$$

$$\begin{aligned}
&\Bigg[\_Cn\,(n+1)\,a(n+1) + \_C0\,(n-1)\,n\,a(n) - 2\_C\left(\sum_{k=0}^{n-1} (k+1)\,a(k+1)\,a(n-1-k) \right) \\
&- 2\_C0\left(\sum_{k=0}^{n-2} (k+1)\,a(k+1)\,a(n-2-k) \right) - \_C\left(\sum_{k=0}^{n} a(k)\,a(n-k) \right) \\
&- \_C0\left(\sum_{k=0}^{n-1} a(k)\,a(n-1-k) \right) + \frac{5n\_C0a(n)}{2} + \frac{5\_C\,(n+1)\,a(n+1)}{2} = 0, \\
&(-\_C0z - \_C)\,y(z)^2 + (-2\_C0\,z^2 - 2\_Cz)\,y(z)\left(\frac{d}{dz}y(z) \right) \\
&+ (\_C0\,z^2 + \_Cz)\left(\frac{d^2}{dz^2}y(z) \right) + \left(\frac{5\_C}{2} + \frac{5\_C0z}{2} \right)\left(\frac{d}{dz}y(z) \right) = 0 \Bigg]
\end{aligned} \tag{9}$$

3

Substituting $\_C = 2$ *and* $\_C0 = 0$ *gives the expected guess. That is equivalent to taking an element of the basis of the* $\mathbb{K}$-*module of all solutions.* ∎

Therefore we obtain the following theorem.

Theorem 3. *The generating function of the sequence defined by*

$$a(n) := \zeta(2n + 2) = (-1)^n (2\pi)^{2n+2} \frac{B_{2n+2}}{2 (2n + 2)!}, \ n \geq 0$$

where B_n *is the* n^{th} *Bernoulli number, satisfies the recurrence equation*

$$(2n + 5)(n + 1) a(n + 1) - 2\left(\sum_{k=0}^{n} 2(k + 1) a(k + 1) a(n - 1 - k) + a(k) a(n - k)\right) = 0, \qquad (10)$$

and, equivalently, its generating function is a solution of the differential equation

$$2z \left(\frac{d^2}{dz^2} y(z)\right) + 5 \frac{d}{dz} y(z) - 4zy(z) \left(\frac{d}{dz} y(z)\right) - 2y(z)^2 = 0. \qquad (11)$$

Proof. The generating function of $(a(n))_{n \geqslant 0}$ can be deduced from the well-known relation

$$-\frac{\pi z}{2} \cot(\pi z) + \frac{1}{2} = \sum_{n=1}^{\infty} \zeta(2n) z^{2n}. \qquad (12)$$

Therefore we get the generating function $\frac{1 - \pi \cot(\pi\sqrt{z})\sqrt{z}}{2z}$ which satisfies (11). □

Example 4. *The up/down or "zig-zag" numbers from* `https://oeis.org/A000111` *have the exponential generating function* $\tan\left(\frac{z}{2} + \frac{\pi}{4}\right)$. *This can be recovered by our implementation. Indeed, the values from OEIS, each divided by the factorial of their index (starting from zero), lead to the following guess.*

$$\left[(n + 1)(n + 2) a(n + 2) - \left(\sum_{k=0}^{n} (k + 1) a(k + 1) a(n - k)\right) = 0, \frac{d^2}{dz^2} y(z) - y(z) \left(\frac{d}{dz} y(z)\right) = 0\right] \qquad (13)$$

Solving the differential equation and matching the first two Taylor coefficients with the first two terms of the sequence give exactly the desired generating function. Note that using `FPS:-FindQRE` *and* `FPS:-QDE`, *one can also compute the same recurrence equation and differential equation from* $\tan\left(\frac{z}{2} + \frac{\pi}{4}\right)$, *respectively.* ∎

References

[1] Jason P Bell, Stefan Gerhold, Martin Klazar, and Florian Luca. 2008. Non-holonomicity of sequences defined via elementary functions. *Annals of Combinatorics* 12, 1 (2008), 1–16.

[2] Jérémy Berthomieu and Jean-Charles Faugère. 2016. Guessing linear recurrence relations of sequence tuples and p-recursive sequences with linear algebra. In *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation*. 95–102.

[3] William A Harris Jr and Yasutaka Sibuya. 1985. The reciprocals of solutions of linear ordinary differential equations. *Advances in Mathematics* 58, 2 (1985), 119–132.

[4] Waldemar Hebisch and Martin Rubey. 2011. Extended rate, more GFUN. *Journal of Symbolic Computation* 46, 8 (2011), 889–903.

[5] Manuel Kauers and Christoph Koutschan. 2022. Guessing with Little Data. *arXiv preprint arXiv:2202.07966* (2022).

[6] Christian Mallinger. 1996. *Algorithmic manipulations and transformations of univariate holonomic functions and sequences.* Master's thesis. Kepler University, Linz.

[7] Mateusz Michałek and Bernd Sturmfels. 2021. *Invitation to Nonlinear Algebra.* Vol. 211. American Mathematical Soc.

[8] LX Châu Ngô and Franz Winkler. 2011. Rational general solutions of parametrizable AODEs. *Publicationes Mathematicae Debrecen* 79, 3-4 (2011), 573–587.

[9] Mutsumi Saito, Bernd Sturmfels, and Nobuki Takayama. 2013. *Gröbner Deformations of Hypergeometric Differential Equations.* Vol. 6. Springer Science & Business Media.

[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] Richard P Stanley. 1980. Differentiably finite power series. *European journal of combinatorics* 1, 2 (1980), 175–188.

[12] Bertrand Teguia Tabuguia. 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`.

[13] Bertrand Teguia Tabuguia and Wolfram Koepf. 2021. On the representation of non-holonomic power series. *arXiv preprint arXiv:2109.09574* (2021).

[14] Doron Zeilberger. 2007. The Holonomic Ansatz II. Automatic Discovery (!) And Proof (!!) of Holonomic Determinant Evaluations. *Annals of Combinatorics* 11, 2 (2007).