Computing the Generating Function of a Series Given Its First Few Terms

François Bergeron and Simon Plouffe

CONTENTS

We outline an approach for the computation of a good candidate for the generating function of a power series for which only the first few coefficients are known. More precisely, if the derivative, the logarithmic derivative, the reversion, or another transformation of a given power series (even with polynomial coefficients) appears to admit a rational generating function, we compute the generating function of the original series by applying the inverse of those transformations to the rational generating function found.

1. INTRODUCTION

We address the problem of finding the generating function $f(x)$ of a power series

$$\alpha(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n + \cdots,$$

of which we know only a limited number of initial terms. We say that $\alpha(x)$ has *precision* n if all coefficients up to x^n are known. Clearly, in the absence of additional information, the knowledge of $\alpha(x)$ to any finite precision is not sufficient to determine $f(x)$ uniquely.

One instance when the problem can be solved is when $f(x)$ is known a priori to be a rational function

$$\frac{p_0 + p_1 x + \cdots + p_j x^j}{q_0 + q_1 x + \cdots + q_k x^k} \quad \text{with } p_j, q_k \neq 0, \tag{1.1}$$

and the precision of $\alpha(x)$ is at least $j + k$. Many good algorithms exist for computing $f(x)$ in this case. A naive one is to use the method of indeterminate coefficients in (1.1), with $j + k = n$. Better algorithms make use of (for example) Padé approximants. The function `convert/ratpoly` provided by the computer algebra system Maple [Char et al. 1985] includes the Padé approximants method.

Bergeron had support from NSERC-Canada and FCAR-Québec.

1058-6458/92 $0.50 per page

If we don't know that the generating function is rational, we can still apply a rational function approximation algorithm to $\alpha(x)$, to obtain an expression of the form (1.1) whose Taylor expansion coincides with $\alpha(x)$ throughout the known terms. If we find out that $k+j$ is much less than the precision n, we can consider the rational fraction obtained a good candidate for the generating function $f(x)$. The greater n is with respect to $j+k$, the more confident we can be in our guess.

Our purpose here is to show that one can easily extend the class of series for which a good candidate for a generating function can explicitly be computed from the knowledge of just enough terms of a series. The main idea is to try to transform the series into one that admits a rational generating function. If this transformation is successful, in the sense that the result appears to be rational, one need only apply the inverse transformation to the resulting rational function in order to produce an explicit candidate for the generating function of the original series. Thus, a measure of rationality for series is crucial to our scheme.

Using this idea, we wrote a Maple program that will find generating functions such as

$$\tan x, \qquad \exp(te^x - t), \qquad (1-4x)^{-3/2},$$
$$\exp\biggl(\frac{1-\sqrt{1-2\,xt}}{x} - t\biggr) \quad \text{and} \quad \frac{1}{1-xe^{A(x)}},$$

where $A(x)$ is the solution to the functional equation $A(x) = x\exp A(x)$—and even more complex ones. The program is described in Section 2, and examples are given in Section 3 that show it to be surprisingly successful. It typically gives results in a few seconds on a Mips/3000 or on a Macintosh IIfx. Moreover, it works with series whose coefficients are polynomials or rational functions, as well as numbers; the generating function in such cases involves a formal parameter, as in the case of $\exp(te^x - t)$ above, which arises in connection with Stirling polynomials of the second kind (see Example 8 in Section 3).

2. THE PROGRAM

The heart of the program is a test for the existence of a good rational function approximation (1.1) for a given series, where *good* is defined to mean that $k+j$ is less than the precision n of the series. This rationality test is implemented in the function `testrat`, which returns either the rational function that has been found, or the keyword `FAIL`.

The power of the program lies in the association of this rationality test with operations such as differentiation, logarithmic differentiation and reversion. (Recall that a series

$$\alpha(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n + \cdots$$

with $a_0 = 0$ and $a_1 \neq 0$ has a unique *reversion* $\alpha^{\langle -1\rangle}(x)$, that is, a series satisfying $\alpha^{\langle -1\rangle}(\alpha(x)) = x$. The generating function of $\alpha^{\langle -1\rangle}(x)$ is inverse to the generating function of $\alpha(x)$, and the first n terms of $\alpha^{\langle -1\rangle}(x)$ depend only on the first n terms of $\alpha(x)$. The *logarithmic derivative* of a series $\alpha(x)$ is $\alpha'(x)/\alpha(x)$.)

In general, the first step of a computation is to execute some transformation Γ on a given series $\alpha(x)$, then to test the resulting series for rationality. If $\Gamma(\alpha(x))$ admits a good rational generating function $f(x)$, the program computes $\Gamma^{-1}(f(x))$, where Γ^{-1} is the transformation inverse to Γ. Note that some operations Γ, such as differentiation, reduce the precision of the series.

This strategy is implemented by calling `testrat` with the functions `testdrat`, `testdlograt` and `testrevrat`. Each of these three functions takes three arguments: the series, the variable (which we have been calling x), and the type of test that should be performed on the transform. The last argument allows tests to be combined: for example, the call `testrevrat(series,x,testdlograt)` will test the logarithmic derivative of the reversion of the series for rationality. These tests, or compositions of them, are successively called by the main program (named `generating` in the examples that follow), which returns a generating function if possible.

Some renormalization of the series is included in `testdrat`, `testdlograt` and `testrevrat`, so that further operations can always be applied. For instance, a series should preferably be of the form

$$x + a_2x^2 + \cdots + a_nx^n + O(x^{n+1}).$$

for reversion.

3. EXAMPLES

The sidebars on this page and the next show a number of representative examples of use of the program `generating`. In some cases, the output has been simplified, using Maple. We use standard mathematical notation for ease of reading, but the Maple input and output is straightforward. The input for Example 1, for example, would be

```
> generating(x + x^2 + 2 x^3 + 3 x^4 +
>     5 x^5 + 8 x^6 + O(x^7));
```

where `>` is the Maple prompt. The program outputs either "`The generating function of this series appears to be ...`" or "`I can find no generating function for this series.`"

Some of the examples were selected from the forthcoming second edition of N. J. A. Sloane's *Handbook of Integer Sequences* [Sloane]. We applied the program to a great number of power series, both ordinary and exponential, corresponding to the sequences in that book (that is, the coefficients of the series were the terms of the sequences). We chose our examples either for their intrinsic elegance, or because they appear to be unknown, or to illustrate the power of the method. Some examples illustrate the use of the program on series with polynomial coefficients.

Example 1. This is the series coming from the Fibonacci sequence. Here `generating` uses directly Maple's function `convert/ratpoly`. The smallest precision for which the result comes out right is six, as shown. With a direct use of this `ratpoly` function (and a simple rejection test) we obtained generating functions for about 600 out of the 4568 sequences in [Sloane].

Example 2. Here the program took the derivative.

Example 3. This is a specialization at $t = -1$ of the next example.

Example 4. This is the exponential generating function for Hermite polynomials. Observe how the input series can have polynomial coefficients, and how the number of terms needed to yield a significant result is quite small.

Example 5. Here the program took the logarithmic derivative.

Example 6. Several generating functions with exponents such as $\frac{3}{2}$, $\frac{5}{2}$, $\frac{7}{2}$ and $\frac{11}{2}$ were obtained when we ran our program on the sequences appearing in [Sloane].

Example 7. This is the exponential generating function for Stirling polynomials of the first kind, which count permutations by number of cycles.

	Input	Output
1	$x + x^2 + 2x^3 + 3x^4 + 5x^5 + 8x^6 + O(x^7)$	$\dfrac{-x}{-1 + x + x^2}$
2	$2 + 5x + \frac{11}{2}x^2 + \frac{19}{3}x^3 + \frac{29}{4}x^4 + \frac{41}{5}x^5 + \frac{55}{6}x^6 + \frac{71}{7}x^7 + \frac{89}{8}x^8 + \frac{109}{9}x^9 + O(x^{10})$	$\dfrac{2 - x^2}{(1-x)^2} + \ln\dfrac{1}{1-x}$
3	$1 + x + x^2 + \frac{2}{3}x^3 + \frac{5}{12}x^4 + \frac{13}{60}x^5 + \frac{19}{180}x^6 + \frac{29}{630}x^7 + \frac{191}{10080}x^8 + \frac{131}{18144}x^9 + O(x^{10})$	$\exp\bigl(x + \frac{1}{2}x^2\bigr)$
4	$1-xt+\bigl(\frac{1}{2}+\frac{1}{2}t^2\bigr)x^2-\bigl(\frac{1}{2}t+\frac{1}{6}t^3\bigr)x^3+\bigl(\frac{1}{8}+\frac{1}{4}t^2+\frac{1}{24}t^4\bigr)x^4-\bigl(\frac{1}{8}t+\frac{1}{12}t^3+\frac{1}{120}t^5\bigr)x^5+O(x^6)$	$\exp\bigl(\frac{1}{2}x(-2t + x)\bigr)$
5	$1 + x + x^2 + \frac{5}{6}x^3 + \frac{17}{24}x^4 + \frac{73}{120}x^5 + \frac{97}{180}x^6 + \frac{2461}{5040}x^7 + \frac{3631}{8064}x^8 + \frac{152531}{362880}x^9 + O(x^{10})$	$\dfrac{\exp\bigl(\frac{1}{4}x^2 + \frac{1}{2}x\bigr)}{\sqrt{1-x}}$
6	$1 + 24x + 270x^2 + 2240x^3 + 15750x^4 + 99792x^5 + 588588x^6 + 3294720x^7 + 17721990x^8 + 92378000x^9 + O(x^{10})$	$\dfrac{1 + 10x + 4x^2}{(1-4x)^{7/2}}$
7	$1 + tx + \frac{1}{2}(t^2 + t)x^2 + \frac{1}{6}(t^3 + 3t^2 + 2t)x^3 + \frac{1}{24}(t^4 + 6t^3 + 11t^2 + 6t)x^4 + \frac{1}{120}(t^5 + 10t^4 + 35t^3 + 50t^2 + 24t)x^5 + O(x^6)$	$\left(\dfrac{1}{1-x}\right)^t$

Example 8. This is the exponential generating function for Stirling polynomials of the second kind, which count partitions of a set by number of parts. This result was obtained through a double logarithmic derivative.

Example 9. This illustrates the use of a rationality test on the reversion of a series. The reversion of this generating function is $x/(1+x)^3$; therefore the generating function $f(x)$ is obtained as the real solution of the cubic equation

$$(1+f(x))^3\, x - f(x) = 0.$$

Example 10. This generating function has two parameters, and admits as one specialization the generating function for Laguerre polynomials. One can find a generating function for most of the classical orthogonal polynomials using our program on the first seven or so terms of their series.

Example 11. This generating function counts functions from a set into itself with weight t^k, where k is the number of recurrent points in the function. $\mathrm{Rev}(f(x),x)$ stands for the inverse for composition of $f(x)$. If we denote by $A(x)$ the solution to the functional equation $A(x) = x\exp(A(x))$, the generating function is equal to

$$\frac{1}{1-txe^{A(x)}}.$$

$A(x)$ is the generating function for rooted trees.

Many other functions such as $\tan x$, $\arctan x$, or $\arcsin x$ also appeared as generating functions in our experiments.

4. CONCLUSIONS

The success of our approach, and also its limitations, depend on the set of transformations tried before a rationality test is made. Many transfor-

	Input	Output
8	$1+tx+\frac{1}{2}(t^2+t)x^2+\frac{1}{6}(t+3t^2+t^3)x^3+\frac{1}{24}(t+7t^2+6t^3+t^4)x^4 + \frac{1}{120}(t+15t^2+25t^3+10t^4+t^5)x^5+O(x^6)$	$\exp(te^x - t)$
9	$x+3x^2+12x^3+55x^4+273x^5+1428x^6 + 7752x^7+43263x^8+246675x^9+O(x^{10})$	$-1+\dfrac{\left(12\sqrt{81x-12}-108\sqrt{x}\right)^{1/3}}{6\sqrt{x}} - \dfrac{\left(12\sqrt{81x-12}+108\sqrt{x}\right)^{1/3}}{6\sqrt{x}}$
10	$1+(t+s)x+\frac{1}{2}(t^2+2ts+s^2+t+2s)x^2 + \frac{1}{6}(t^3+3t^2s+3ts^2+s^3+3t^2+9ts+6s^2+2t+6s)x^3 + \frac{1}{24}(t^4+4t^3s+6t^2s^2+4ts^3+s^4+6t^3+24t^2s + 30ts^2+12s^3+11t^2+44ts+36s^2+6t+24s)x^4 + \frac{1}{120}(t^5+5t^4s+10t^3s^2+10t^2s^3+5ts^4+s^5+10t^4 + 50t^3s+90t^2s^2+70ts^3+20s^4+35t^3+175t^2s + 260ts^2+120s^3+50t^2+250ts+240s^2+24t+120s)x^5 + O(x^6)$	$\left(\dfrac{1}{1-x}\right)^t \exp\left(\dfrac{sx}{1-x}\right)$
11	$xt+\left(t+t^2\right)x^2+\left(\frac{3}{2}t+2t^2+t^3\right)x^3 + \left(4t^2+3t^3+\frac{8}{3}t+t^4\right)x^4+\left(\frac{25}{3}t^2+\frac{15}{2}t^3+\frac{125}{24}t+4t^4+t^5\right)x^5 + \left(18t^2+18t^3+\frac{54}{5}t+12t^4+5t^5+t^6\right)x^6 + \left(\frac{343}{8}t^3+\frac{98}{3}t^4+\frac{2401}{60}t^2+\frac{35}{2}t^5+\frac{16807}{720}t+6t^6+t^7\right)x^7 + \left(\frac{16384}{315}t+7t^7+t^8+24t^6+\frac{160}{3}t^5+\frac{256}{3}t^4+\frac{512}{5}t^3+\frac{4096}{45}t^2\right)x^8 + O(x^9)$	$t\,\mathrm{Rev}\left(\dfrac{x}{xt+1}\exp\left(-\dfrac{x}{xt+1}\right), x\right)$

mations beyond differentiation, logarithmic differentiation and reversion may be considered. For instance, one could choose any invertible function $f(x)$ and consider the following transformations on a series $\alpha(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n + O(x^{n+1})$:

$$\theta_f(\alpha(x)) = \text{taylor}(a_0 + a_1 f(x) + \cdots + a_n f(x)^n),$$
$$\theta^f(\alpha(x)) = \text{taylor}(f(a_1 x + a_2 x^2 + \cdots + a_n x^n)).$$

Here taylor(g) stands for the operation of taking the Taylor expansion around 0 of a function g, and θ^f is defined when $a_0 = 0$. If $f^{\langle -1\rangle}$ denotes the reversion of $f(x)$, one easily checks that

$$(\theta_f)^{-1}(g(x)) = g(f^{\langle -1\rangle}(x)),$$
$$(\theta^f)^{-1}(g(x)) = f^{\langle -1\rangle}(g(x)).$$

One nice case is when $f(x) = \ln x$ in θ_f. This transformation allows the computation of generating functions that are rational functions of the exponential. For instance, one could obtain in this manner the generating function

$$\frac{e^x - 1}{2 - e^x}$$

for the series

$$x + \tfrac{3}{2}x^2 + \tfrac{13}{6}x^3 + \tfrac{25}{8}x^4 + \tfrac{541}{120}x^5 + \tfrac{1561}{240}x^6 + \tfrac{47293}{5040}x^7 + \tfrac{36389}{2688}x^8 + \tfrac{7087261}{362880}x^9 + O(x^{10}),$$

which is the exponential series for ordered partitions of a set. As it happens, our program found this generating function by other means, namely by taking the derivative of the reversion of the series, whose generating function is

$$\frac{1}{(1+2x)(1+x)}.$$

To describe other possible extensions of our approach, we recall some definitions. A series $y(x)$, with coefficients in $\mathbf{K}$, is said to be *differentiably finite* or *D-finite* [Stanley 1980] if it satisfies some nontrivial linear differential equation

$$p_0(x)y + p_1(x)y' + \cdots + p_k(x)y^{(k)} = 0 \qquad (4.1)$$

with coefficients $p_j(x) \in \mathbf{K}[x]$. A series $y = y(x)$ is said to be *constructible differentially finite* or *CDF* [Bergeron and Reutenauer 1990] if, for some $k \geq 1$, there exist k series $y_1, \ldots, y_k$, with $y_1 = y$, and polynomials $P_1, \ldots, P_k$ with coefficients in $\mathbf{K}$, satisfying

$$y_i' = P_i(y_1, \ldots, y_k) \quad \text{for } i = 1, \ldots, k. \qquad (4.2)$$

Both of these classes of series contain polynomials, algebraic series, and the Taylor expansion around 0 of usual functions such as e^x, $\log(1+x)$, or the trigonometric functions. They are also closed under addition and multiplication, and under composition with algebraic series. However, the CDF class is not closed under Hadamard (termwise) product, whereas the D-finite class is. On the other hand, CDF is closed under differentiation, integration, inversion $(1/y(x))$, composition and reversion.

Neither class is contained in the other. All CDF series are analytic around 0, so $\sum_n n!\,x^n$ is not CDF, though it is D-finite. On the other hand, the series expansion around 0 of $1/\cos x$ is not D-finite, but is CDF.

Both classes allow for the characterization of a wide range of generating functions. If one knows the form of the liner differential equation (4.1) or the system (4.2)—that is, the number of equations and the degrees of the polynomials—the exact equation or system characterizing a given series or a set of series can then be found from the series' first terms. In the case of D-finite series, this technique has already been proposed and implemented by Guttmann [Brak and Guttmann 1990]. For CDF series, we have an experimental program that has been used to obtain nice new generating functions such as

$$F(u,v,x) = \frac{\alpha^2}{e^x\big((1+u)\sin(\tfrac{1}{2}\alpha x) - \cos(\tfrac{1}{2}\alpha x)\big)^2}, \qquad (4.3)$$

where $\alpha = \sqrt{2v - (1+u)^2}$. This is a generating function (with parameters) for the number of maximal up-going paths in the composition poset (ongoing research in collaboration with S. Dulucq and M. Bousquet-Mélou). Function (4.3) is not D-finite but is CDF. To obtain it, we used the first few terms of the series

$$\begin{aligned}1 &+ ux + \tfrac{1}{2}(v+u^2)x^2 + \tfrac{1}{6}(v + 4vu + u^3)x^3 \\ &+ \tfrac{1}{24}(v + 4v^2 + 6vu + 11vu^2 + u^4)x^4 \\ &+ \tfrac{1}{120}(v + 14v^2 + 34uv^2 \\ &\qquad + 8vu + 23u^2v + 26vu^3 + u^5)x^5 \\ &+ \cdots,\end{aligned}$$

obtained by explicit enumeration of the objects considered, in order to find the system

$$F' = F(1+G), \qquad F(u,v,0)=1,$$
$$G' = v + (1+u)G + G^2/2, \qquad G(u,v,0)=0.$$

Expression (4.3) is easily computed from this.

Our first implementation of `generating` computed a generating function for either the ordinary or the exponential series of about 1000 out of the 4568 sequences appearing in [Sloane]. Since the first version of this article was written, a Maple package implementing some ideas presented here, as well as others such as the D-finite approach, has been written by Bruno Salvy and Paul Zimmermann of INRIA [Salvy and Zimmermann]. It is now available as a shared package under the name "`gfun`". (To learn more about obtaining shared packages, type `?share` to Maple.) The analogue of our function `generating` in `gfun` is the function `guessgf`. Giving `guessgf` the right set of options results in its using the set of transformations described in Section 2 of this paper.

ACKNOWLEDGEMENTS

We would like to thank G. Labelle, N. J. A. Sloane and an anonymous referee for their constructive comments.

REFERENCES

[Bergeron and Reutenauer 1990] F. Bergeron and C. Reutenauer, "Combinatorial resolution of systems of differential equations, III: A special class of differentially algebraic series", *Europ. J. Combin.* **11** (1990), 501–512.

[Brak and Guttmann 1990] R. Brak and A. J. Guttmann, "Algebraic approximants: a new method of series analysis", *J. Phys.* A**23** (1990), L1331–1337.

[Char et al. 1985] Bruce W. Char et al., *Maple User's Guide*, 4th ed., Watcom Publications, Waterloo, Ont., 1985.

[Salvy and Zimmermann] B. Salvy and P. Zimmermann, "GFUN: A Maple Package for the Manipulation of Generating and Holonomic Functions in one Variable", to appear in *ACM Trans. in Math. Software*.

[Sloane] N. J. A. Sloane, *A Handbook of Integer Sequences*, 2nd ed., Academic Press, to appear (1st ed., 1973).

[Stanley 1980] R. P. Stanley, "Differentiably finite power series", *Europ. J. Combin.* **1** (1980), 175–188.

[Zeilberger 1991] D. Zeilberger, "A Maple program for proving hypergeometric identities", *SIGSAM Bulletin* **25**(3) (1991), 4–13.

François Bergeron, LACIM, Université du Québec à Montréal, Montréal H3C 3P8, Canada
Spring 1993: LaBRI, Université de Bordeaux I, 33405 Talence, France (bergeron@catalan.math.uqam.ca)

Simon Plouffe, LACIM, Université du Québec à Montréal, Montréal H3C 3P8, Canada

Received October 30, 1991; revised June 11, 1992