Symbolic Asymptotics: Multiseries of Inverse Functions

Bruno Salvy, John Shackell

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Symbolic Asymptotics: Multiseries of Inverse Functions

Bruno SALVY, John SHACKELL

N ° 3264
Septembre 1997

THÈME 2

Rapport de recherche

ISSN 0249-6399

Symbolic Asymptotics: Multiseries of Inverse Functions

Bruno SALVY, John SHACKELL

Thème 2 — Génie logiciel
et calcul symbolique
Projet Algo

Rapport de recherche n° 3264 — Septembre 1997 — 23 pages

Abstract: We give an algorithm to compute an asymptotic expansion of multiseries type for the inverse of any given exp-log function. An example of the use of this algorithm to compute asymptotic expansions in combinatorics via the saddle-point method is then treated in detail.

(Résumé : tsvp)

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : (33) 01 39 63 55 11 – Télécopie : (33) 01 39 63 53 30

Asymptotique automatique : multiséries d'inverses fonctionnels

Résumé : Nous donnons un algorithme calculant un développement asymptotique du type multisérie pour l'inverse de toute fonction exp-log. Un exemple d'application de cet algorithme au calcul de développements asymptotiques en combinatoire par la méthode du col est traité en détail.

Symbolic Asymptotics: Multiseries of Inverse Functions

BRUNO SALVY AND JOHN SHACKELL

Inria Rocquencourt, France
University of Kent at Canterbury, U. K.

(26 September 1997)

We give an algorithm to compute an asymptotic expansion of multiseries type for the inverse of any given exp-log function. An example of the use of this algorithm to compute asymptotic expansions in combinatorics via the saddle-point method is then treated in detail.

Introduction

This article is part of a series on the symbolic computational aspects of asymptotics. While computer algebra has encountered great success in areas like symbolic integration and linear differential equations, the handling of asymptotics was long a source of difficulty. For many years the general systems such as Maple, Macsyma and Reduce used a collection of *ad hoc* techniques generally based on compositions of limits or l'Hôpital's rule and later on series or generalized series expansions (Geddes and Gonnet 1989). A more systematic treatment requires automating the determination of the proper asymptotic scale for a specific computation and dealing with the indefinite cancellation problem, exemplified by $\exp(x^{-1}+e^{-x})-\exp(x^{-1})$ as $x \to +\infty$. If one tries to expand naively the two exponential series, the terms in x^{-1} dominate the first expansion and perpetually cancel with the corresponding terms of the second.

The automation of asymptotics began in the mid-to-late eighties. Work of Hardy (1910) emphasizes the importance in asymptotics of the class of exp-log functions (functions obtained from a variable x and the set of rational numbers $\mathbb{Q}$ by closure under field operations and the applications of exp and $\log|.|$). One of the first effective results in this area is an algorithm given by Shackell (1990) which computes the limit of any exp-log function. This was developed and implemented in Maple by Gruntz (1996). An earlier package for asymptotic computation formed part of the $\Lambda\Upsilon\Omega$ system (Flajolet *et al.* 1991, Salvy 1991a, 1991b). The basic methods of Shackell (1990) have been extended to allow other functions in the signature by Shackell (1995, 1996).

In this article, we consider the asymptotics of inverse functions in a computer-algebra setting. The asymptotics of inverse functions were, of course, studied well before the development of electronic computers, but they proved troublesome. For example Hardy (1911) states as a conjecture that there exist exp-log functions whose inverse is not asymptotically equivalent to an exp-log function. This conjecture was only proved recently (Shackell 1993a, Van den Dries *et al.* 1997, Van der Hoeven 1997). We gave an algorithm for functional inversion of exp-log functions in terms of *nested expansions* in (Salvy and Shackell 1992). For instance, given the input

$$ye^{\log^2 y\, e^{\sqrt{\log\log y}}} = x, \qquad x \to +\infty,$$

this algorithm produced the output

$$y = \exp\left[\frac{\sqrt{\log x}}{e^{\frac{\sqrt{2}}{4}\sqrt{\log\log x}}} \cdot e^{1/8} \cdot \left(1 - \frac{\sqrt{2}\log\log^{-1/2} x}{64} + \frac{\log\log^{-1} x}{4096} + \frac{383\sqrt{2}}{786432}\log\log^{-3/2} x - \frac{1535}{100663296}\log\log^{-2} x + \cdots\right)\right]. \tag{0.1}$$

In the present article, we develop an algorithm which produces another kind of expansion in the form of *multiseries* (precise definitions are given in Section 1). These are close to the traditional definition of asymptotic expansions but can provide a finer estimate. For instance, in the example above, by setting parameters of our new algorithm, the output can be either as in (0.1) or one of the following (successive) refinements:

$$\exp(e^U)\left[1 - \frac{2e^{-\sqrt{U}}}{\sqrt{U}+4} + \frac{2e^{-2\sqrt{U}}}{(\sqrt{U}+4)^2} - \frac{4}{3}\frac{e^{-3\sqrt{U}}}{(\sqrt{U}+4)^3} + O(e^{-4\sqrt{U}})\right],$$

$$\exp(e^U)\exp[-\frac{2e^{-\sqrt{U}}}{\sqrt{U}+4}]\left[1 + \frac{8 - 2U^{-1/2} - U^{-1} + U^{-3/2}}{(4+U^{-1/2})^3}e^{-U-2\sqrt{U}} + O(e^{-2U})\right],$$

with

$$U = \frac{\log\log x}{2} - \frac{1}{8}\sqrt{8\log\log x + 1} + \frac{1}{8},$$

defined as the inverse function of $2y(x) + y(x)^{1/2}$ composed with $\log\log x$.

Multiseries were introduced in (Van der Hoeven 1997, Richardson *et al.* 1996). Other names for these kinds of asymptotic expansions or very similar ones are *hyperasymptotics* (Berry and Howls 1990), *exponential asymptotics* (Meyer 1980), *asymptotics beyond all orders* (Costin and Kruskal 1996). They are also closely related to the *transseries* of Écalle (1992), of which they can be viewed as an effective version. Multiseries seem to have some advantages over nested expansions especially in the way in which results are presented, though we would claim that nested expansions also have advantages; in particular they are canonical and they often make it easier to develop and prove algorithms. In (Richardson *et al.* 1996) an algorithm was given to compute multiseries for the class of exp-log functions. The present paper can be viewed as the natural next step. While this paper was being written, the thesis of J. Van der Hoeven (1997) appeared. It contains a short section on functional inversion, following on from (Van der Hoeven 1994). The standpoint is similar to ours, but our algorithm is different and we give much more detail. Van der Hoeven (1997) also considers problems of much greater generality. It would appear that here the author currently relies on the use of algebraic differential equations for zero-equivalence testing, (Shackell 1993c, Péladan-Germa 1995). Our treatment avoids this in most cases.

In Bourbaki (1961) (see also Dieudonné (1968)), it is shown that if g is an exp-log function such that $g(x)/x \to 0$, then one can obtain a recurrence for the inverse of $x - g(x)$

by setting $u_0(x) = x$ and $u_n(x) = x + g(u_{n-1}(x))$ for $n \geq 1$. By using substitutions, one can then obtain the inverse of any exp-log function Ψ which can be written in the form $\Psi = \psi - g$ where ψ is an exp-log function whose inverse is an exp-log function and $g = o(\psi)$. Of course these results still leave a number of problems. It is not clear how one can find the decomposition $\Psi = \psi - g$ or even whether one always exists. Even in the basic case where $\psi(x) = x$, the expansion obtained is in terms of g, but methods for expanding g in an appropriate scale did not exist at the time when (Bourbaki 1961) appeared. This approach was applied to general functional inversion in a transseries context by Écalle (1992). However as pointed out in (Van der Hoeven 1997), Écalle's formula can give wrong answers if applied directly to the transseries to be inverted. Since transseries are formal objects, the question of whether they give asymptotic formulæ for inverse functions does not arise. In this paper, we use an iteration derived from Écalle's formula and prove that it yields an algorithm for giving asymptotic series for inverse functions.

In Section 1, we give our algorithm to invert multiseries of exp-log functions. We also show how to handle expressions built from elementary functions and a single inverse function. This is a non-trivial extension, since there may be cancellation between the inverse function and other subexpressions. It might be thought that these cancellation problems only occur in specially contrived examples. However the last section of this paper is concerned with an application of the saddle-point method to some problems in combinatorics, and here such cancellations are to be expected. Moreover it is necessary to use multiseries rather than straightforward asymptotic series. For example, we show that in one very natural example the answer will be wrong by a factor tending to infinity if the ordinary asymptotic series expansions are used without care. All our examples are computed using a pilot implementation we have developed in Maple.

Acknowledgement. This work was supported in part by the Long Term Research Project Alcom-IT (#20244) of the European Union, and also by the Alliance Programme.

1. Algorithm

1.1. Definitions

Multiseries are in effect multivariate power series in which the powers may be non-integral, but must tend to infinity, and the variables are elements of a *scale*. We now give more precise definitions for these notions, very similar to those in (Richardson *et al.* 1996).

Definition 1.1. *An* asymptotic scale *is a finite ordered set* $\{t_1, \ldots, t_m\}$ *of positive exp-log functions tending to zero such that* $\log t_i = o(\log t_{i+1})$*, for* $i = 1, \ldots, m-1$.

The condition on the scale elements implies that they are mutually transcendental.

Definition 1.2. *A* multiseries expansion *of a function* f *with respect to a one-element scale* $\{t_1\}$ *is an asymptotic series*

$$f = \sum_{\alpha \in S} a_\alpha t_1^\alpha,$$

where the a_α*'s are constant and* $S = \alpha_1\mathbb{N} + \alpha_2\mathbb{N} + \cdots + \alpha_k\mathbb{N} + \beta$ *is a finitely generated set of exponents with real positive* α_i*'s and* $\beta \in \mathbb{R}$.

For $m > 1$, a multiseries expansion *of a function f with respect to the scale $\{t_1, \ldots, t_m\}$ is an asymptotic series of the form*

$$f = \sum_{\alpha \in S} f_\alpha t_m^\alpha,$$

where again S is a finitely generated set of exponents and each f_α is an exp-log function having a multiseries expansion with respect to the scale $\{t_1, \ldots, t_{m-1}\}$.

Thus multiseries are asymptotic series in the scale element of fastest decrease, with coefficients which are non-zero functions having multiseries in the remaining scale elements. From the point of view of symbolic computation, it is important that we have finite expressions for these coefficient functions and for any coefficients in *their* expansions etc., in order that zero-equivalence tests can be made. We denote by $T(f)$ and $t(f)$ the respective scale elements of greatest and least decrease which actually occur in the multiseries of f. More formally, we note that one part of the algorithm of (Richardson *et al.* 1996) expresses f as analytic function of the scale elements, $f = A(t_1, \ldots, t_m)$.We can then define $T(f) = t_k$ and $t(f) = t_i$ by the conditions $\partial A/\partial t_k \neq 0$, $\partial A/\partial t_i \neq 0$ and $\partial A/\partial t_j = 0$ for $j > k$ and for $j < i$. This means in particular that if $m > k$, the multiseries for f with respect to $\{t_1, \ldots, t_m\}$ if of the form $f = f_0 t_m^0$. Note that $T(f)$ and $t(f)$ are not necessarily the elements of greatest and least decrease in the scale. For example we might have $T(f) = l_1^{-1}(x)$ and $t(f) = \exp(-l_5^2(x))$, while the scale would contain x^{-1} and $l_5^{-1}(x)$. Here we used some notation that will be employed throughout the paper. We write l_r for the r-times iterated logarithm, and similarly e_r for the iterated exponential.

We also want any scale we use to satisfy further properties.

Definition 1.3. *We say that a scale $\{t_1, \ldots, t_m\}$ is* complete *if the following conditions hold:*

1. *Each t_i is either the reciprocal of an iterated logarithm, $l_{r_i}^{-1}(x)$, or an exponential, $\exp(-h_i)$, where h_i and h_i' have multiseries expansions with respect to $\{t_1, \ldots, t_{i-1}\}$;*
2. *x^{-1} is an element of the scale and if $t_i = l_{r_i}^{-1}(x)$ with $r_i > 0$, then $l_{r_i-1}^{-1}(x)$ is also an element of the scale.*

Thus elements of complete scales are either reciprocals of iterated logarithms, or exponentials of functions possessing multiseries in the smaller elements of the scale. Throughout this paper, our scales will be complete and our multiseries will be computable with finite expressions for their coefficients. We recall that the main result of (Richardson *et al.* 1996) is an algorithm to compute a complete scale and a multiseries expansion in this scale for any given exp-log function. Arithmetical operations with multiseries having the same scale are just the usual ones for asymptotic series, except that we need to take care regarding the closed forms for the coefficients. We refer the reader to (Richardson *et al.* 1996) for the details here.

1.2. Description of the algorithm

Let $f(x)$ be a function tending to infinity which has a multiseries expansion in a (complete) scale $\{t_1(x), \ldots, t_m(x)\}$. We want to calculate a multiseries for the inverse function $y(x) = f^{(-1)}(x)$. The algorithm consists of three parts which we now describe along with an example. Proofs are given in Section 2.

1.2.1. Exact computation

This part starts from an exp-log function $f(x)$ as above. Let $t_k = T(f)$. We first compute the leading term $f_0(x) = c(x)t_k^{\alpha_0}(x)$ of $f(x)$ with respect to $t_k(x)$ and we let $g = f - f_0$ be the tail. We note that since $f \to \infty$, we cannot have $\alpha_0 > 0$, and thus we have the following breakdown into cases:
Case 1. $\alpha_0 = 0$ *and* $\log t_k = O(\log x)$.
Here we compute recursively the inverse, Y, of $f(e^x)$; the result is $\exp(Y(x))$.
Case 2. $\alpha_0 = 0$ *and* $\log x = o(\log c(x))$ *or* $\alpha_0 < 0$ *and* $T(f) \neq t(f)$.
We compute recursively the inverse, Y, of $\log f$; the result is $y(x) = Y(\log x)$;
Case 3. $\alpha_0 = 0$, $\log f_0 = O(\log x)$ *and* $\log x = o(\log g)$ *or* $\alpha_0 < 0$ *and* $T(f) = t(f)$.
In these cases we use the method given in the next section to compute $y(x)$ from the equation

$$y[x + g(y_0(x))] = y_0(x) \qquad \text{where} \qquad f_0(y_0(x)) = y_0(f_0(x)) = x. \tag{1.1}$$

We show in Lemma 2.6 that we arrive at Case 3 after a finite number of steps.

In order to illustrate the rôle of this part of our algorithm we consider the problem of inverting the following function:

$$f^{(0)}(x) = x^2 e^x + 1.$$

The algorithm from (Richardson *et al.* 1996) readily computes the scale $\{t_1 = 1/x, t_2 = e^{-x}\}$ and the decomposition

$$f^{(0)} = f_0^{(0)} + g^{(0)} \qquad \text{with} \quad f_0^{(0)} = t_1^{-2}t_2^{-1}, \quad g^{(0)} = 1.$$

Then the algorithm is invoked recursively with input $f^{(1)} = \log f$. The scale is found to be $\{t_1 = 1/\log x, t_2 = 1/x, t_3 = e^{-x}\}$, $f^{(1)}$ is rewritten as

$$f^{(1)} = \log f^{(0)} = 2\log x + x + \log(1 + x^{-2}e^{-x})$$

and the following decomposition is obtained

$$f^{(1)} = f_0^{(1)} + g^{(1)} \qquad \text{with} \quad f_0^{(1)} = 2/t_1 + 1/t_2, \quad g^{(1)} = \log(1 + t_2^2 t_3).$$

Now $\alpha_0 = 0$ and $T(f^{(1)}) = t_3$ The next part of the algorithm is then called with input

$$y[x + \log(1 + y_0^{-2}(x)e^{-y_0(x)})] = y_0(x), \qquad y_0(x) + 2\log y_0(x) = x. \tag{1.2}$$

The same treatment applies to $f^{(2)}(x) = x + 2\log x$ whose inverse is defined in the last equation. One first gets the scale $\{t_1 = 1/\log x, t_2 = 1/x\}$ and the obvious decomposition

$$f^{(2)} = f_0^{(2)} + g^{(2)} \qquad \text{with} \quad f_0^{(2)} = 1/t_2, \quad g^{(2)} = 2/t_1.$$

Since we are in Case 2, this leads to a recursive invocation of the algorithm with input $f^{(3)} = \log f^{(2)}$. The scale is unchanged, and $f^{(3)}$ is rewritten

$$f^{(3)}(x) = \log f^{(2)}(x) = \log x + \log(1 + 2\log x/x).$$

Again we are in Case 2 and the algorithm is again called recursively with input $f^{(4)}(x) =$

$f^{(3)}(e^x)$. The scale is now $\{t_1 = 1/x, t_2 = e^{-x}\}$, and at last the next part of the algorithm (given in the next section) can be called with input

$$y_2[x + \log(1 + 2y_3(x)e^{-y_3(x)})] = y_3(x), \qquad y_3(x) = x. \tag{1.3}$$

To summarize, we obtain the following exact representation for the inverse of $x^2e^x + 1$:

$$\begin{cases} Y(x) = y(\log x), & y \text{ inverse of } 2\log x + x + \log(1 + e^{-x}/x^2),\\ y[x + \log(1 + y_0^{-2}(x)e^{-y_0(x)})] = y_0(x), & y_0 \text{ inverse of } x + 2\log x,\\ y_0(x) = y_1(\log x), & y_1 \text{ inverse of } \log x + \log(1 + 2\log x/x),\\ y_1(x) = \exp(y_2(x)), & y_2 \text{ inverse of } x + \log(1 + 2xe^{-x}),\\ y_2[x + \log(1 + 2y_3(x)e^{-y_3(x)})] = y_3(x), & y_3 \text{ inverse of } x. \end{cases}$$

1.2.2. Iteration

This part starts from $f_0(x)$, $g(x)$ and (1.1). The result is a truncation (in that we only compute a finite number of terms) of the multiseries expansion

$$y = \sum_{i \geq 0} c_i(y_0)t_k^i(y_0)$$

in the scale $\{t_1(y_0), \ldots, t_m(y_0)\}$, the c_i's being explicitly computed exp-log functions.

Following Écalle (1992), we define an operator K by

$$K(h) = h[x + g(y_0(x))] - h(x). \tag{1.4}$$

Then (1.1) may be rewritten as

$$(I + K)y(x) = y_0(x), \tag{1.5}$$

where I denotes the identity, from which it is natural to expect

$$y(x) = (I - K + K^2 - K^3 + \cdots)y_0(x). \tag{1.6}$$

This leads to consideration of the following iteration due to Écalle (1992)

$$u_{n+1}(x) = u_n(x) + (-1)^n K^n(y_0(x)), \qquad 0 \leq n \leq N, \tag{1.7}$$

where $u_0(x) = y_0(x)$ and N is the number of desired terms in the multiseries expansion.

After the computations of the previous section, either $\alpha_0 < 0$ and $T(f) = t(f)$ or $\alpha_0 = 0$, $\log f_0 = O(\log x)$ and $\log x = o(\log g)$. In the former case, the iteration (1.7) can be performed by power series manipulations and creates no difficulty. We now describe the iteration in the latter case. Then, $g(y_0)$ has a multiseries expansion in $\{t_1(y_0), \ldots, t_m(y_0)\}$ starting with a positive power of $t_k(y_0)$ (the name of the variable—x or y_0—is of no consequence). The steps are as follows:

1. Compute the N first derivatives of $y_0(x)$ in terms of y_0 via $y_0' = 1/f_0'(y_0)$.
2. Deduce the (truncated) multiseries expansion of $y_0(x + g(y_0))$ with respect to $t_k(y_0)$ from these derivatives and the formula

$$y_0[x + g(y_0)] = \sum_{n \geq 0} y_0^{(n)}(x)\frac{g^n(y_0)}{n!}.$$

3. For $i = 1, \ldots, k$, compute multiseries expansions for the $t_i[y_0(x + g(y_0))]$ in the scale $\{t_1(y_0), \ldots, t_m(y_0)\}$ as follows.

(a) For values of j from 1 to the maximal number of iterated logarithms in the scale, compute the multiseries expansion of the corresponding $t_i[y_0(x+g(y_0))]$ with respect to $t_k(y_0)$ using the previous value, the formula

$$\log(l_{j-1}(y_0(x+g(y_0)))) = l_j(y_0) + \log\left(1 + \frac{l_{j-1}(y_0(x+g(y_0))) - l_{j-1}(y_0)}{l_{j-1}(y_0)}\right)$$

and the classical series expansion for $\log(1+u)$;

(b) For i from 1 to k and for those t_i's which are exponentials, say $t_i = \exp(-h_i)$, compute the corresponding multiseries expansion of $t_i[y_0(x+g(y_0))]$ using

$$\exp(-h_i[y_0(x+g(y_0))]) = t_i(y_0)\exp\{h_i(y_0(x)) - h_i[y_0(x+g(y_0))]\}$$

, where $h_i[y_0(x+g(y_0))]$ in the right-hand side is first expanded by replacing $t_j(y_0)$ in the multiseries expansion for $h_i(y_0)$ by the multiseries for $t_j(y_0+g(y_0))$ for $j = 1,\dots,i-1$.

4. Starting from $\phi_1(y_0) = y_0$ and using (1.4), the iteration (1.7) is then performed efficiently by:

(a) setting $\phi_{i+1}(y_0) = \phi_i[y_0(x+g(y_0))] - \phi_i(y_0)$, the expansion of the first summand being computed by replacing the t_i's by their expansion computed in Step 3;

(b) returning the expansion with respect to $t_k(y_0)$ of

$$\sum_{i=1}^{N}(-1)^{i+1}\phi_i(y_0).$$

In the same example as before, say with $N=2$, we start from (1.2) and first obtain

$$y_0' = \frac{1}{1+2/y_0}, \qquad y_0'' = \frac{2}{y_0^2(1+2/y_0)^3}.$$

Here the scale is $\{t_1(y_0) = 1/\log y_0,\ t_2(y_0) = 1/y_0,\ t_3(y_0) = T(y_0) = \exp(-y_0)\}$. From our expressions for the derivatives of y_0 it follows that

$$\begin{aligned} 1/t_2(y_0(x+g)) &= y_0(x+\log(1+y_0^{-2}e^{-y_0})) \\ &= y_0 + \frac{e^{-y_0}}{y_0^2(1+2/y_0)} - \frac{1+4/y_0+2/y_0^2}{2y_0^4(1+2/y_0)^3}e^{-2y_0} + O(e^{-3y_0}). \end{aligned} \tag{1.8}$$

Step 3 then computes

$$\begin{aligned} t_1(y_0(x+g)) &= 1/\log[y_0(x+\log(1+e^{-2y_0}/y_0^2))] \\ &= \frac{1}{\log y_0} - \frac{e^{-y_0}}{y_0^3(1+2/y_0)} + \frac{1+5/y_0+4/y_0^2}{2y_0^5(1+2/y_0)^3}e^{-2y_0} + O(e^{-3y_0}), \end{aligned}$$

and

$$\begin{aligned} t_3(y_0(x+g)) &= \exp[-y_0(x+\log(1+y_0^{-2}e^{-y_0}))] \\ &= e^{-y_0} - \frac{e^{-2y_0}}{y_0^2(1+2/y_0)} + \frac{1+3/y_0+1/y_0^2}{y_0^4(1+2/y_0)^3}e^{-3y_0} + O(e^{-4y_0}). \end{aligned}$$

Now the ϕ_is are very easy to compute:

$$\begin{aligned}
\phi_1 &= y_0 = 1/t_2,\\
\phi_2 &= \phi_1(y_0(x+g)) - \phi_1 = \frac{t_3 t_2^2}{(1+2t_2)} - \frac{1+4t_2+2t_2^2}{2(1+2t_2)^3} t_2^4 t_3^2 + O(t_3^3),\\
\phi_3 &= \phi_2(y_0(x+g)) - \phi_2 = -\frac{1+4t_2+2t_2^2}{(1+2t_2)^3} t_2^4 t_3^2 + O(t_3^3),
\end{aligned}$$

where for ϕ_2 we used (1.8). Whence the final result:

$$y = y_0 - \frac{e^{-y_0}}{y_0^2(1+2/y_0)} - \frac{1+4/y_0+2/y_0^2}{2y_0^4(1+2/y_0)^3} e^{-2y_0} + O(e^{-3y_0}). \tag{1.9}$$

The original function $Y(x)$ is then recovered as $y(\log x)$.

A similar treatment applies to (1.3), and leads to

$$\begin{aligned}
x + \log(1+2xe^{-x}) &= 1/t_1(y_3(x+g)) = x + 2xe^{-x} - 2x^2e^{-2x} + O(x^3e^{-3x}),\\
\exp[-x+\log(1+2xe^{-x})] &= t_2(y_3(x+g)) \quad = e^{-x} - 2xe^{-2x} + 4x^2e^{-3x} + O(x^3e^{-4x}).
\end{aligned}$$

The iteration produces

$$\phi_1 = x, \qquad \phi_2 = 2xe^{-x} - 2x^2e^{-2x} + O(x^3e^{-3x}), \qquad \phi_3 = -4(x^2-x)e^{-2x} + O(x^3e^{-3x}).$$

Hence

$$y_2(x) = x - 2xe^{-x} - 2(x^2-2x)e^{-2x} + O(x^3e^{-3x}). \tag{1.10}$$

Approximating momentarily $y(x)$ by $y_0(x)$, the above expansion induces the estimate $\exp(y_2(\log\log x))$ for the original function $Y(x)$. So

$$\begin{aligned}
Y(x) &= \exp\left[\log\log x - 2\frac{\log\log x}{\log x} - 2\frac{\log\log^2 x + 2\log\log x}{\log^2 x} + O\left(\frac{\log\log^3 x}{\log^3 x}\right)\right]\\
&= \log x - 2\log\log x + 4\frac{\log\log x}{\log x} + 4\frac{\log\log^2 x - 2\log\log x}{\log^2 x} + O\left(\frac{\log\log^3 x}{\log^3 x}\right).
\end{aligned} \tag{1.11}$$

This expansion is further refined by (1.9).

1.2.3. Substitution

The result of the previous part is a multiseries expansion of y in terms of y_0, which is itself an inverse function. We now consider the problem of obtaining a multiseries expansion for y in terms of x. It is a consequence of Liouville's theorem that in general this cannot be expected with base elements and coefficients which are exp-log functions. For instance, let f be an exp-log function whose inverse y is not asymptotic to an exp-log function (see Shackell (1993a)), then $\log f$ has an inverse $\exp(y)$ which cannot have a multiseries expansion since otherwise its logarithm would be asymptotic to an exp-log function.

In many cases however, it is possible to produce a multiseries expansion for y in terms of x from those of y in terms of y_0, and y_0 in terms of x, based on the following algorithm for substitution, which is in the same vein as Step 3 above.

(a) For values of j from 1 to the maximal number of iterated logarithms in the scale, we compute the multiseries for $l_j(y_0)$ by taking the logarithm of previous multiseries.

(b) For i from 1 to k and for those t_i which are exponentials, say $t_i = \exp h_i$ where h_i has a multiseries in $t_1, \dots, t_{i-1}$, substitute the multiseries $t_1(y_0(x)), \dots, t_{i-1}(y_0(x))$ into the multiseries for h_i and exponentiate the result.

The source of difficulty is the exponentiation which may require an extension of the scale by a function which is not exp-log.

It is however possible to proceed with a scale containing functions which are not exp-log. In this case, what we have are exp-log expressions for the coefficients in terms of different variables, $y_0, y_1, \dots, y_{k-1}, x$. Hence we need to be able to test for zero equivalence of expressions of the form $F(y_0, \dots, y_k, x)$, where F is an exp-log function. Theoretically at least, this problem can be solved, modulo an oracle for constants, using differential equations. For each of the y_i's, and any given exp-log functions of them, satisfy differential equations over the constants, and one of the known differential-equations methods can therefore be used, (Shackell 1993c, Péladan-Germa 1995).

In practice however, unless f is particularly simple, this theoretical algorithm is likely to be impossibly slow. We have sought a better method, which uses the structure of the inverses in a more efficient way, but except in the case when $k \leq 1$ we have not yet succeeded in finding one. When $k = 1$, we only have to contend with exp-log functions of x and y_0, which we can rewrite as exp-log functions of y_0 using the relation $x = f_0(y_0(x))$; so any of the methods for exp-log functions (Rothstein and Caviness 1979, Shackell 1989, Shackell 1993c, Péladan-Germa 1995) can be used. (See Section 3 for an example of this situation).

In the example above, the simple substitution algorithm suffices. We start from (1.9), which is expressed in the scale $\{t_1 = 1/y_0, t_2 = e^{-y_0}\}$. We also have the multiseries expansion, (1.10), of $y_2(x) = \log y_0(e^x)$ in the scale $\{1/x, e^{-x}\}$.

From that we deduce

$$y_1(x) = e^{y_2(x)} = e^x - 2x + 4xe^{-x} + O(x^2e^{-2x}),$$
$$e^{y_1(x)} = e^{e^{y_2(x)}} = e^{e^x}\left[e^{-2x} + 4xe^{-3x} + O(x^2e^{-4x})\right].$$

Hence

$$y_0(x) = y_1(\log x) = x - 2\log x + 4\log x/x + O(\log^2 x x^{-2}),$$
$$e^{y_0(x)} = e^x\left[x^{-2} + 4\log x/x^3 + O(\log^2 x x^{-4})\right].$$

These expansions make it possible to compute the multiseries expansion of any coefficient of (1.9), hence of $Y(x) = y(\log x)$, in the scale $\{1/\log\log x, 1/\log x, 1/x\}$. For instance (1.11) corresponds to the first term, y_0, in (1.9). Our computation gives the next term as

$$Y(x) - y_0(\log x) = -\frac{1}{x}\left[1 - \frac{2}{\log x} + 4\frac{\log\log x - 1}{\log^2 x} + O\left(\frac{\log\log x}{\log^3 x}\right)\right].$$

2. Iteration theorem and proof of the algorithm

Formula (1.6) was given by Écalle (1992). He also stated

$$K(h) = h(x + g(y_0)) - h(x) = \sum_{r\geq 1} \frac{h^{(r)}(x)}{r!}(g(y_0))^r. \tag{2.1}$$

However Écalle was concerned with formal series, and it should be stressed that so far (1.6) and (2.1) are only valid in this sense. In order to obtain a multiseries in our sense we have to prove that the top-level series is indeed an asymptotic series for the function $y(x)$. We have to show how to obtain expressions for the coefficient functions, in a form to which zero-equivalence tests may be applied. Similarly we need to be sure that the series we have for these coefficient functions are indeed asymptotic series for them, and that we have suitable expressions for their coefficients, and so on.

One of the main purposes of this section is to prove the following iteration theorem.

THEOREM 2.1. *Let $f(x)$ be an exp-log function with $f \to \infty$ as $x \to \infty$ and let $S = \{t_1(x), \ldots, t_m(x)\}$ be a complete scale for f, with $t_k = T(f)$. Suppose that $\log x = o(\log(t_k(x)))$, that the leading term of the multiseries expansion of f with respect to S is of the form $f = f_0 t_k^0$ and that $\log f_0 = O(\log x)$. Write $g = f - f_0$, and let y be the inverse of f and y_0 the inverse of f_0. Define the sequence of functions $\{u_n(x)\}$ by $u_0(x) = y_0(x)$ and*

$$u_{n+1}(x) = y_0(x) - Ku_n(x) = y_0(x) - [u_n(x + g(y_0(x))) - u_n(x)], \quad n \geq 0.$$

Then $\lim(y(x) - u_n(x)) = 0$ and more precisely

$$y(x) - u_n(x) \sim g^{n+1}(y_0)\psi_n(x),$$

where $\log(\psi_n(x)) = o((g(y_0)))$.

We note that the computations of Subsection 1.2.1 reduce us to the case covered by this theorem. Our proof is based on results in Hardy fields, and we begin by recalling the properties of these that we need.

2.1. HARDY FIELDS

Let $\mathcal{X}$ be the ring of germs at $+\infty$ of $\mathcal{C}^\infty$ functions. So elements of $\mathcal{X}$ are represented by functions defined on intervals of the form (a, ∞), and two functions define the same germ if they are identical on such an interval. We shall often blur the distinction between functions and germs where this is harmless. A *Hardy field* is then defined to be a subring of $\mathcal{X}$ which is a field closed under differentiation. The germs of exp-log functions form a Hardy field (Rosenlicht 1983a). The name comes form the slightly larger field of L-functions studied by Hardy.

The relevance of the definition to asymptotics is perhaps not immediately apparent. However non-zero elements of Hardy fields have to possess multiplicative inverses, and thus cannot have arbitrarily large zeros. Therefore they are ultimately positive or ultimately negative. The same must be true of their derivatives, and so elements of Hardy fields are ultimately monotonic. Hence they tend to limits, which can be infinite. Moreover a total order can be defined on any Hardy field by setting $f > g$ whenever $f(x) > g(x)$ for x sufficiently large. We shall make frequent use of this order on the Hardy fields that we meet. The fact that elements can be compared, together with the existence of limits and the closure under differentiation, makes the theory of Hardy fields an extremely useful tool in a number of areas of asymptotics.

The first such result can be found for example in (Bourbaki 1961, V.22, Prop. 7). It makes it possible to compare derivatives in a way that is not possible for arbitrary $\mathcal{C}^\infty$ functions.

LEMMA 2.1. *Let f and g be two elements of a Hardy field such that g does not tend to a non-zero finite constant then $f = o(g)$ (resp. $f \sim g$) implies $f' = o(g')$ (resp. $f' \sim g'$).*

Next we need to introduce comparability classes. Two elements of a Hardy field, f and g which tend to infinity are said to be *comparable* if each is dominated by a power of the other; that is to say if there exist positive integers m and n such that $f < g^m$ and $g < f^n$. The relation is extended to other Hardy-field elements by declaring $\pm f$ and $\pm f^{-1}$ to all be comparable to each other, and closing under transitivity. Finally any two elements which tend to non-zero finite limits are regarded as comparable. Comparability is then an equivalence relation on the non-zero elements of the Hardy field. We write $\gamma(f)$ for the equivalence class of f, and refer to it as the *comparability class* of f. The asymptotic ordering between functions can be carried over to comparability classes by writing $\gamma(f) < \gamma(g)$ when f and g both tend to infinity and $f^n < g$ for all positive integers n. In other words, $\gamma(f) < \gamma(g)$ is a short way to write $\log|f| = o(\log|g|)$. Thus for example

$$\gamma(l_2(x)) < \gamma(l_1(x)) < \gamma(x) < \gamma(\exp(\log^2(x))) < \gamma(e_1(x)) < \gamma(e_2(x)) \cdots .$$

We shall need the following three lemmas. A proof of the first can be found, for example, in (Shackell 1996).

LEMMA 2.2. *Let h be an element of a Hardy field with h not asymptotic to a non-zero constant. Then $\gamma(h') \leq \max\{\gamma(x), \gamma(h)\}$, with equality when $\log|h| \not\sim \log x$.*

The second lemma is given in Bourbaki (1961), Proposition 4. The statement there is in terms of exp-log functions, but the proof goes over easily to the more general situation.

LEMMA 2.3. *Let f and g be two elements of a Hardy field, and suppose that $gf'/f \to 0$ and that $g/x \to 0$. Then $f(x + g(x)) \sim f(x)$.*

The result of our last lemma appears in different guises in many places (for example (Boshernitzan 1981)). The version we give is taken from (Salvy and Shackell 1992).

LEMMA 2.4. *Let f be an element of a Hardy field which tends to infinity. Then the inverse function of f belongs to a Hardy field.*

Although this lemma is obviously important for us, it does not allow us to conclude that all the objects arising in our computations are Hardy-field elements. For it is not generally the case that the union of two Hardy fields is contained in a Hardy field, (Boshernitzan 1987). So for example, an expression containing two different inverse functions might not lie in any Hardy field. In fact, since all the functions we use are obtained from exp-log functions, it follows from the result of (Wilkie 1996) that they will be Hardy-field elements. However, in order to keep our proofs elementary, we have chosen not to use the power of Wilkie's theorem. This has meant that we have had to take extra care when estimating derivatives, for example, and our proofs are probably a little longer at times as a result.

For a more detailed study of Hardy fields, the reader is referred to to the literature, Bourbaki (1961), Robinson (1972), Boshernitzan (1981,1982,1986,1987), Rosenlicht (1983a,1983b,1984,1987), Shackell (1993b).

2.2. Proof of the iteration theorem

We write $G = g \circ y_o$. Let $\{u_n\}$ be defined by $u_0(x) = y_0(x)$, and for $n \geq 0$

$$u_{n+1}(x) = y_0(x) + u_n(x) - u_n(x + G(x)). \tag{2.2}$$

Note that by (1.5), this gives

$$y - u_{n+1} = y - (1 + K)y - Ku_n = -K(y - u_n). \tag{2.3}$$

Proposition 2.1. *For all $k, n \in \mathbb{N}$,*

$$y^{(k)} - u_n^{(k)} = G^{n+1}\psi_{k,n}, \tag{2.4}$$

where $\log|\psi_{0.k}^{(j)}| = o(\log|G|)$ for all $j \in \mathbb{N}$.

Proof. We begin by noting some properties of G. Firstly since g is an exp-log function and y_0 belongs to a Hardy field, it follows from Theorem 1 of (Rosenlicht 1983a) that the Hardy field containing y_0 may be extended to include G. We denote this Hardy field by $\mathcal{F}$. We also recall that $\gamma(G) > \gamma(y_0) \geq \gamma(x)$. Next, by Lemma 2.2, $\gamma(G'/G) \leq \gamma(\log|G|) < \gamma(G)$. So $\log|G'| \sim \log|G|$. Iteration of this gives

$$\log|G^{(j)}| \sim \log|G|, \tag{2.5}$$

for all $j \geq 0$. We note that $x + G = f \circ y_0$, and hence that the inverse function of $x + G$ is $f_0 \circ y$. Then from Lemma 2.3 we have

$$G(x + G) \sim G(x) \sim G(f_0(y(x))), \tag{2.6}$$

the second relation being obtained by substituting $x \mapsto f_0(y(x))$ in the first.

The following result will be needed for the case $n = 0$.

Lemma 2.5.

$$y_0^{(k)}(x) = y^{(k)}(x + G(x)) \cdot (1 + G')^k + \Gamma_k, \tag{2.7}$$

where Γ_k is a polynomial in $G', \ldots, G^{(k)}, y'(x + G), \ldots, y^{(k-1)}(x + G)$ in which every monomial contains a derivative of G.

Proof. We use induction on k. For the case $k = 0$ we may take $\Gamma_0 = 0$, since $y_0(x) = y(f(y_0(x))) = y(x + G)$.

Then for the induction step, we have

$$\begin{aligned} y_0^{(k+1)}(x) &= \left(y^{(k)}(x + G) \cdot (1 + G')^k + \Gamma_k(x)\right)' \\ &= y^{(k+1)}(x + G) \cdot (1 + G')^{k+1} + ky^{(k)}(x + G) \cdot (1 + G')^{k-1}G'' + \Gamma_k', \end{aligned}$$

and we see that we may take $\Gamma_{k+1} = \Gamma_k' + y^{(k)}(x + G) \cdot k(1 + G')^{k-1}G''$. This establishes Lemma 2.5. □

A simple induction then shows that $\mathcal{F}$ contains $y^{(k)}(x+G)$ for all k. Next we note that $\gamma(y^{(k)}(x + G)) < \gamma(G)$ for all $k \in \mathbb{N}$. To see this, we first observe that $y^{(k)}(x + G) \sim y_0^{(k)}$ by (2.7). Then unless some derivative of y_0 is asymptotic to a non-zero constant, Lemma 2.2 shows that $\gamma(y_0^{(k)}) \leq \gamma(y_0) < \gamma(G)$. To cover the possibility that y_0 is asymptotic to a constant multiple of x^M for some $M \in \mathbb{N}$ with $M < k-1$, we use the inductive hypothesis that y_0 has a multiseries in a basis whose elements have comparability class less than G.

Now we are in a position to establish (2.4) in the case $n = 0$. From (2.7) we have

$$\begin{aligned} y^{(k)}(x) - y_0^{(k)}(x) &= y^{(k)}(x) - y^{(k)}(x+G)\cdot(1+G')^k - \Gamma_k \\ &= -y^{(k+1)}(x+\theta G)\cdot G + y^{(k)}(x+G)\cdot(1-(1+G')^k) - \Gamma_k, \end{aligned} \tag{2.8}$$

where $0 < \theta(x) < 1$. From Lemma 2.5 and (2.5), it is clear that

$$\gamma\left(\{y^{(k)}(x+G)(1-(1+G')^k) - \Gamma_k\}/G\right) < \gamma(G),$$

and by Lemma 2.2, the same holds for any derivative of $\{y^{(k)}(x+G)(1-(1+G')^k)-\Gamma_k\}/G$. Moreover for any j, $y^{(k+j)}(x+\theta G)$ lies between $y^{(k+j)}(x)$ and $y^{(k+j)}(x+G)$. We have already seen that $\gamma(y^{(k+j)}(x+G)) < \gamma(G)$. To obtain the corresponding conclusion for $y^{(k+j)}(x)$ (bearing in mind that we do not assume that $y \in \mathcal{F}$), we observe that for every $\varepsilon \in \mathbb{R}^+$

$$|y^{(k+j)}(x)| = |y^{(k+j)}((x+G)\circ f_0(y(x)))| < |G(f_0(y(x))|^{-\varepsilon},$$

which gives the desired result since $G(f_0(y(x))) \sim G(x)$ by (2.6). It now follows from (2.8) that

$$y^{(k)}(x) - y_0^{(k)}(x) = G\psi_{0,k}(x)$$

with $\log|\psi_{0.k}^{(j)}| = o(\log|G|)$ for all j. This completes the proof of Proposition 2.1 in the case when $n = 0$.

The Induction on n

We are supposing that for all $k \in \mathbb{N}$, $y^{(k)} - u_n^{(k)} = G^{n+1}\psi_{k,n}$, where $\log|\psi_{k,n}^{(j)}| = o(\log|G|)$ for all j. We have to prove the corresponding formula with n replaced by $n+1$. When $k = 0$ we have from (2.3) that $y - u_{n+1} = -K(y-u_n) = -Kh$, with $h = y - u_n$. Thus

$$\begin{aligned} y - u_{n+1} = h(x+G) - h(x) &= h'(x+\theta G)\cdot G \qquad (0 < \theta < 1) \\ &= \{y'(x+\theta G) - u_n'(x+\theta G)\}\, G. \end{aligned} \tag{2.9}$$

By the induction hypothesis

$$y'(x+\theta G) - u_n'(x+\theta G) = G^{n+1}(x+\theta G)\psi_{1,n}(x+\theta G) \sim G^{n+1}(x)\psi_{1,n}(x+\theta G), \tag{2.10}$$

since $G(x+\theta G)$ lies between $G(x)$ and $G(x+G)$ and (2.6) applies. Also

$$\log|\psi_{1,n}^{(j)}(x+\theta G)| = o(\log|G(x+\theta G)|) = o(\log|G(x)|),$$

for all j. Hence from (2.9) and (2.10)

$$y(x) - u_{n+1}(x) = G^{n+2}(x)\psi_{0,n+1}(x)$$

with $\psi_{0,n+1}(x) = \psi_{1,n}(x+\theta G)G^{n+1}(x+\theta G)/G^{n+1}(x)$; we see that $\psi_{0,n+1}(x)$ has the required property.

Now suppose that we have our conclusion for k and we want to prove it for $k+1$. On differentiating (2.4), we obtain

$$\begin{aligned} y^{(k+1)} - u_n^{(k+1)} &= (n+1)G^n G' \psi_{n,k} + G^{n+1}\psi'_{n,k} \\ &= G^{n+1}\left\{(n+1)\frac{G'}{G}\psi_{n,k} + \psi'_{n,k}\right\}. \end{aligned}$$

We therefore take $\psi_{n,k+1} = G'\psi_{n,k}/G + \psi'_{n,k}$, and we see that $\psi_{n,k+1}$ satisfies our conditions, since $\gamma(G'/G) < \gamma(G)$ by (2.5).

By induction, (2.4) holds for all n and k, and so we have proved Proposition 2.1 and Theorem 2.1. □

Corollary 2.1. *Under the conditions of Theorem 2.1, our algorithm produces a multiseries for $y(x)$ in the scale $\{t_1(y_0(x)), \ldots, t_m(y_0(x))\}$.*

Proof. Since $\gamma(g(y_0)) = \gamma(t_k(y_0))$, the Iteration Theorem implies that the multiseries for $y(x)$ in the scale $\{t_1(y_0(x)), \ldots, t_m(y_0(x))\}$ up to $t_k^n(y_0)$ is the same as the multiseries for $u_n(x)$. The conclusion of the theorem then follows from noting that the expansions produced by the iteration part of our algorithm are in powers of $t_k(y_0(x))$ with coefficients that are all exp-log functions $\Phi(y_0)$ such that $\gamma(T(\Phi)) < \gamma(t_k)$. □

Corollary 2.2. *Suppose that the conditions of Theorem 2.1 hold. Then for any scale S containing $\{t_1, \ldots, t_m\}$, the output of our algorithm is the multiseries expansion of y with respect to S.*

The meaning of this corollary is that there does not exist a more refined expansion of y than the one produced by our algorithm.

Proof. We first consider the case $S = \{t_1, \ldots, t_{m+1}\}$ where t_{m+1} is such that $\gamma(t_{m+1}) > \gamma(t_m)$. Suppose that the multiseries for y is of the form $\tilde{y} + \eta$ where $\tilde{y}$ is the multiseries computed by our algorithm and η is a term of the same comparability class as $t_{m+1}(y_0)$. Theorem 2.1 shows that if we substitute the multiseries for f into the multiseries for $\tilde{y}$, we get a multiseries with the single term x in the scale $\{t_1(y_0(f)), \ldots, t_m(y_0(f))\}$. Then the multiseries for $y(f)$ in the larger scale S would be of the form $x + \phi$, with $\gamma(\phi) = \gamma(t_{m+1}(y_0))$. But that cannot be so since as a function $y(f)$ is identically equal to x. Thus we have shown that the scale with have obtained for f^{-1} cannot contain an element whose comparability class is larger than that of $t_m(y_0)$.

Now the same argument applies inductively to y_0 and shows that there cannot be a different multiseries expansion than the one we have obtained with respect to a scale with extra elements. □

2.3. Termination and correctness of the algorithm

While convergence of the iteration has been proved, we still have to show that the exact-computation stage converges and that the coefficients of our multiseries are exp-log functions of y_0.

Lemma 2.6. *The computations of Section 1.2.1 terminate.*

The recursive steps occur in cases 1 and 2. Then if $T(f) \neq t(f)$, taking the logarithm

$$\log f = \alpha_0 \log t_k + \log c(x) + \log\left(1 + \frac{t_k^{-\alpha_0} g}{c(x)}\right) \tag{2.11}$$

makes the new α_0 equal to 0, because $\gamma(\log T(f)) < \gamma(T(f))$ (and $T(f)$ still occurs in the new g). Now, if α_0 is equal to zero, this remains the case after changing x into e^x, while $\gamma(t_k)$ is increased. After finitely many steps $\gamma(t_k) > \gamma(x)$. Then if $\gamma(c(x)) > \gamma(x)$, equation (2.11) with $\alpha_0 = 0$ shows that taking the logarithm keeps α_0 equal to 0 while reducing $\gamma(c(x))$. After finitely many steps $\gamma(c(x)) \leq \gamma(x)$ and the recursion stops.

We now turn to the Iteration part of the algorithm and show that the coefficients c_i's indeed are exp-log functions.

In Step 1, the derivatives of y_0 are expressed in terms of exp-log functions which do not involve $g(y_0)$ (or equivalently $t_k(y_0)$) but only smaller comparability classes. Moreover, g being itself an exp-log function, the multiseries expansions of $g^n(y_0)$ with respect to $t_k(y_0)$ has exp-log function coefficients. Therefore the multiseries for $y_0[x + g(y_0)]$ constructed in Step 2 has exp-log coefficients. Step 3 is designed in such a way that only power series expansions with respect to $t_k(y_0)$ are performed and these preserve the exp-log character of the coefficients. Step 4 then relies on Step 3 to perform the iteration using the previous expansions. The desired conclusion then follows. We summarize our results so far as follows.

THEOREM 2.2. *Let $f(x)$ be an exp-log function with $f \to \infty$ as $x \to \infty$. Suppose that $S = \{t_1(x), \ldots, t_m(x)\}$ is a complete scale for f and let $t_k = T(f)$. By using the transformations given in Section 1.2.1 we may reduce in a finite number of steps to the case covered by Theorem 2.1.Then our algorithm produces a multiseries expansion of y with respect to $\{t_1(y_0), \ldots, t_m(y_0)\}$ of the form*

$$y = \sum_{i=0}^{\infty} c_i(y_0) t_k^i(y_0), \tag{2.12}$$

where the c_i's are exp-log functions.

3. An application in combinatorics

We now show how the techniques described in this article to deal with inverse functions apply to the computation of the asymptotic behaviour of combinatorial parameters like the average number of parts in the partition of a set with n elements, or its variance, and similar problems.

Let $S = \{1, \ldots, n\}$ be a set of n distinguishable elements. A partition of S is a set of non-empty subsets S_i, $i = 1, \ldots, k$ (called the *parts*) which are mutually disjoint and whose union is equal to S. The number of distinct partitions of S is called the *Bell number* B_n. For instance $B_3 = 5$ because $\{1,2,3\} = \{1\} \cup \{2,3\} = \{2\} \cup \{1,3\} = \{3\} \cup \{1,2\} = \{1\} \cup \{2\} \cup \{3\}$ are the five ways of partitioning $\{1,2,3\}$. Classical combinatorial arguments show that the Stirling number of the second kind $S_{n,k}$ which is the number of partitions of a set of n elements into k parts has the following generating function:

$$\mathcal{B}(u, z) := \sum_{n \geq 0, k \geq 0} S_{n,k} u^k \frac{z^n}{n!} = \exp[u(e^z - 1)],$$

and naturally one has $B_n = \sum_k S_{n,k}$.

3.1. The saddle-point method

We first concentrate on the Bell numbers themselves, with generating function $B(z) = \mathcal{B}(1, z)$. The traditional way to compute their asymptotic expansions is to apply *the saddle-point method* (see (De Bruijn 1981)) to the integral representation

$$\frac{B_n}{n!} = \frac{1}{2i\pi} \oint \frac{B(z)}{z^{n+1}}\, dz.$$

In this representation the contour is any simple loop enclosing the origin. To simplify the notation, define $h(z)$ to be the logarithm of the integrand. The idea is to move the contour to pass through the saddle-point, which is the solution of $h'(R) = 0$, or equivalently of

$$R\frac{B'(R)}{B(R)} - 1 = n. \tag{3.1}$$

Then the integral is concentrated in the neighbourhood of this point. Locally it behaves like a Gaussian integral, and the first order estimate obtained by this method is

$$\frac{B_n}{n!} \sim \frac{\exp(h(R))}{\sqrt{2\pi h''(R)}}. \tag{3.2}$$

A theorem due to Hayman (1956) describes a large class of functions $B(z)$ for which R above is real positive and the formal method just outlined is guaranteed to produce the right asymptotic estimate.

In our examples below, we use a version of this theorem due to Harris and Schoenfeld (1968) and Odlyzko and Richmond (1985) which applies to a smaller class of functions, but yields a full asymptotic expansion instead of first order asymptotics.

In the case of the Bell numbers, the saddle-point equation reads

$$Re^R - 1 = n, \tag{3.3}$$

from which our algorithm retrieves the classical expansion

$$R = \log n - \log\log n + \frac{\log\log n}{\log n} + \frac{1}{2}\frac{\log\log n(\log\log n - 2)}{\log^2 n} + \cdots \tag{3.4}$$

(Fast ways of computing this expansion to a large order are described in (Comtet 1970, Salvy 1994)). This expansion can then be substituted into (3.2) to get the order of growth of the numbers of partitions of a set. Substitution of the expansion of R into $h(R) = \log B(R) - (n+1)\log R$, and into the derivatives $h'(R), h''(R)$, requires handling expansions involving R and n simultaneously. Following the idea of Section 1.2.3, we first replace n by the left-hand side of (3.3) in these expressions. Thus we obtain an exp-log expression in R for the estimate (3.2). Using the algorithm for exp-log functions from (Richardson *et al.* 1996), we get a rewriting of this expression in the scale

$$\ln(R), \quad R, \quad e^R, \quad \exp(e^R), \quad \exp(\ln(R)Re^R),$$

from which we can for instance extract the following multiseries for the estimate;

$$\exp(-\ln(R)Re^R)\exp(e^R)e^{-R/2}\frac{e^{-1}}{\sqrt{2\pi}}\left[1 - \frac{1}{2R} + \frac{3}{8R^2} + O(R^{-3})\right]. \tag{3.5}$$

The last element of the scale being present in the leading term of this estimate, a direct substitution of (3.4) requires an extension of the scale in n, which obscures the result. Instead, we consider the logarithm of the expression above, for which we get the expansion

$$-\ln(R)Re^R + e^R - R/2 + O(1). \tag{3.6}$$

We then substitute the expansion (3.4) into this and get the classical (De Bruijn 1981, p. 108)

$$\ln\frac{B_n}{n!} \sim n\left(-\log\log n + \frac{\log\log n + 1}{\log n} + \frac{1}{2}\frac{\log\log^2 n}{\log^2 n} + \cdots\right);$$

more terms can be obtained by increasing the order of the computations. This expansion follows entirely from the leading term (3.2) and does not require the refinements of (Harris and Schoenfeld 1968, Odlyzko and Richmond 1985).

3.2. The average

The main difficulty in the practical use of the saddle-point method is that the estimate (3.2) is in terms of n and R, while R is only known asymptotically as the inverse of an exp-log function, through (3.1). It is then generally difficult to work with these expansions and handle the asymptotic cancellations that occur.

Our purpose is now to show that in this example and similar ones where the saddle-point method applies, some computations with these expansions are possible thanks to our methods, by working in the right asymptotic scale. The computations alternate between asymptotic and exact representations, replacing n by the left-hand side of (3.1) to get expressions in terms of R only, on which the exp-log machinery can be used. We now illustrate this idea on the average and variance of the number of parts in a partition (or equivalently the Stirling numbers of the second kind).

We apply the same method as above to the generating function

$$C(z) = \left.\frac{\partial\mathcal{B}(z,u)}{\partial u}\right|_{u=1},$$

which has the property that $[z^n]C(z)n!/B_n$ is[†] the average number of parts in a partition of a set of size n. The saddle-point equation is now

$$\frac{R_1e^{2R_1} - e^{R_1} + 1}{e^{R_1} - 1} = n. \tag{3.7}$$

Proceeding as above, we first compute the asymptotic behaviour of R_1. Using any traditional method, we get for R_1 the same estimate (3.2) as for R. However, it turns out that there is a small difference between R and R_1 which is hidden behind an indefinite cancellation, and that this exponentially small difference has an impact on the first order estimate of the average we are looking for! In order to get a more precise idea of the difference between both saddle-points, the idea is to compute both expansions in a finer

† We use the classical notation $[z^n]f(z)$ to denote the nth Taylor coefficient of $f(z)$ at the origin.

scale. Using our algorithm we thus compute the next level of the multiseries, which yields

$$R = \zeta + \frac{e^{-\zeta}}{\zeta+1} - \frac{1}{2}\frac{\zeta+2}{(\zeta+1)^3}e^{-2\zeta} + O(e^{-3\zeta}),$$
$$R_1 = \zeta - \frac{\zeta-1}{\zeta+1}e^{-\zeta} - \frac{1}{2}\frac{3\zeta^3+2\zeta^2-\zeta+4}{(\zeta+1)^3}e^{-2\zeta} + O(e^{-3\zeta}).$$

where ζ is defined as an inverse function by

$$\zeta + \ln(\zeta) = \ln(n).$$

By using this equation we can overcome the problems of indefinite cancellation and estimate the difference between the two saddle-points.

$$\begin{aligned} R - R_1 &= \frac{e^{-\zeta}}{1+1/\zeta} + O(e^{-2\zeta}), \\ &= \frac{1}{n}\left[\log n - \log\log n - 1 + \frac{\log\log n + 1}{\log n} + O(\frac{\log\log^2 n}{\log^2 n})\right]. \end{aligned}$$

When we want to compute the asymptotic behaviour of $[z^n]C(z)/(B_n/n!)$, we shall be faced with an estimate for the denominator in terms of R and an estimate for the numerator in terms of R_1. Since we have zero-equivalence problems when more than one inverse function is present, we shall replace R and R_1 by their expansions in terms of ζ and proceed in this scale with only one inverse function.

Replacing n by the left-hand side of (3.7) in the saddle-point estimate (3.2), where now $h(z)$ is defined as $\log C(z) - (n+1)\log z$, and expressing the estimate in terms of ζ, yields

$$\begin{aligned} [z^n]C(z) = \exp&\left(-\ln(\zeta)\zeta e^{\zeta} + e^{\zeta} + \frac{1}{2}\zeta - \ln(\zeta)\right) \times \\ &\frac{e^{-1}}{\sqrt{2\pi(1+1/\zeta)}}\left[1 - \frac{1}{24}\frac{26+69/\zeta+76/\zeta^2+30/\zeta^3+2/\zeta^4}{(1+1/\zeta)^3}e^{-\zeta} + O(e^{-2\zeta})\right]. \end{aligned}$$

Similarly, when we substitute the expansion of R in terms of ζ in (3.5) we get

$$\begin{aligned} \frac{B_n}{n!} = \exp&\left(-\ln(\zeta)\zeta e^{\zeta} + e^{\zeta} - \frac{1}{2}\zeta - \ln(\zeta)\right) \times \\ &\frac{e^{-1}}{\sqrt{2\pi(1+1/\zeta)}}\left[1 - \frac{1}{24}\frac{2+9/\zeta+16/\zeta^2+6/\zeta^3+2/\zeta^4}{(1+1/\zeta)^3}e^{-\zeta} + O(e^{-2\zeta})\right]. \end{aligned}$$

Thus we get the average number of parts in a partition of a set of n elements is

$$\begin{aligned} \mu_n &= e^{\zeta}\left[1 - \frac{1}{2}\frac{2+3/\zeta+2/\zeta^2}{(1+1/\zeta)^2}e^{-\zeta} + O(e^{-2\zeta})\right], \\ &= \frac{n}{\log n}\left[1 + \frac{\log\log n}{\log n} + \frac{\log\log^2 n - \log\log n}{\log^2 n} + O(\frac{1}{\log^3 n})\right]. \end{aligned}$$

The second estimate is obtained by substituting the expansion of ζ in terms of n (which is (3.4)), into the first one. Actually, the result depends only on the leading term of the first estimate. The answer obtained is in accordance with the first order estimate given by Bender and Richmond (1996) or Sachkov (1995), but interestingly enough it differs by a factor $e = \exp(1)$ from the estimate given by another reference on the subject, (Harper

1967). Also, these references only give the first order estimate. All this gives an idea of the complexity of these calculations when performed by hand.

Note that although they are very close one to the other, if one uses R instead of R_1, the error on μ_n is a factor $\exp(\zeta \ln(\zeta)) \sim \exp[\log n \log\log n - \log\log^2 n - \log n]$, which tends to infinity.

3.3. The variance and other applications

In these computations, the calculation of the variance often leads to further cancellation. It is given by

$$\sigma_n = \frac{[z^n]D(z)}{[z^n]B(z)} - \mu_n^2,$$

where

$$D(z) = \frac{\partial}{\partial u} u \frac{\partial}{\partial u} B(z,u)\bigg|_{u=1}.$$

As above, this computation involves a new saddle-point defined by

$$R_2 \frac{e^{2R_2} + e^{R_2} - 1}{e^{R_2} - 1} - 1 = n.$$

We compute in the same scale as previously, which leads to the asymptotic estimate

$$R_2 = \zeta - \frac{2\zeta - 1}{\zeta + 1} e^{-\zeta} - \frac{3}{2} \frac{2\zeta^3 - 3\zeta + 2}{(\zeta+1)^3} e^{-2\zeta} + O(e^{-3\zeta}).$$

From this we compute the estimate for $[z^n]D(z)$ as before, and get

$$\frac{[z^n]D(z)}{[z^n]B(z)} = e^{2\zeta} - \frac{2 + 2/\zeta + 1/\zeta^2}{(1+1/\zeta)^2} e^{\zeta} - \frac{1}{12} \frac{1/\zeta(10 + 19/\zeta + 22/\zeta^2 - 2/\zeta^3)}{(1+1/\zeta)^5} + O(e^{-\zeta}).$$

In this scale, it is clear that only the leading terms cancel, and there is no indefinite cancellation. So for the variance we get

$$\begin{aligned} \sigma_n &= \frac{e^{\zeta}}{\zeta+1} - \frac{1}{2} \frac{2 + 8/\zeta + 11/\zeta^2 + 9/\zeta^3 + 2/\zeta^4}{(1+1/\zeta)^4} + O(e^{-\zeta}), \\ &= \frac{n}{\log^2 n} \left[1 + \frac{2\log\log n - 1}{\log n} + \frac{3\log\log^2 n - 5\log\log n + 1}{\log^2 n} + O(\frac{\log\log^3 n}{\log^3 n}) \right]. \end{aligned}$$

This is in agreement with (Bender and Richmond 1996) and (Sachkov 1995), who give only the first term. Obviously, our method produces as many terms as desired and gives the higher moments without any difficulty.

To illustrate our algorithm further, we now turn to the number of partitions of a set into partitions. For instance, $\{1,2,3\}$ can be partitioned in 12 ways: three partitions of the type $\{\{1\}\} \cup \{\{2,3\}\}$, three of the type $\{\{1\}\}, \{\{2\},\{3\}\}$, the partition $\{\{1\}\} \cup \{\{2\}\} \cup \{\{3\}\}$ and a partition consisting of one set for each partition of $\{1,2,3\}$. These objects and their further generalizations are studied in statistics, where they are used to model classification hierarchies. The bivariate generating function is now

$$\mathcal{H}(u,z) = \exp[u(\exp(e^z - 1) - 1)].$$

By the same method, we get the average

$$\mu_n = \frac{n}{\log n \log\log\log n}\left[1+\frac{\log\log n+\log\log\log n}{\log n}+O(\frac{\log\log\log n}{\log n\log\log\log n})\right],$$

and the variance

$$\sigma_n = \frac{n}{\log^2 n\ \log\log n}\left[1+\frac{1}{\log\log n}+O(\frac{1}{\log\log^2 n})\right].$$

To the best of our knowledge, these results are new, in large part because a computation by hand would be formidable.

References

Bender, E. A., Richmond, L. B. (1996). Admissible functions and asymptotics for labelled structures by number of components. *Electronic Journal of Combinatorics*, 3(R34).

Berry, M. V., Howls, C. J. (1990). Hyperasymptotics. *Proceedings of the Royal Society. London. Series A.*, 430(1880):653–668.

Boshernitzan, M. (1981). An extension of Hardy's class L of "Orders of infinity". *Journal d'Analyse Mathématique*, 39:235–255.

Boshernitzan, M. (1982). New "Orders of infinity". *Journal d'Analyse Mathématique*, 41:130–167.

Boshernitzan, M. (1986). Hardy fields and existence of transexponential functions. *Aequationes Mathematicae*, 30:258–280.

Boshernitzan, M. (1987). Second order differential equations over Hardy fields. *Journal of the London Mathematical Society*, 35(2):109–120.

Bourbaki, N. (1961). *Éléments de Mathématiques*, chapter V: Fonctions d'une variable réelle (appendice), pages 36–55. Hermann, Paris, 2nd edition.

Comtet, L. (1970). Inversion de $y^\alpha e^y$ et $y\log^\alpha y$ au moyen des nombres de Stirling. *Comptes-Rendus de l'Académie des Sciences*, 270:1085–1088.

Costin, O., Kruskal, M. D. (1996). Optimal uniform estimates and rigorous asymptotics beyond all orders for a class of ordinary differential equations. *Proceedings of the Royal Society. London. Series A.*, 452(1948):1057–1085.

De Bruijn, N. G. (1981). *Asymptotic Methods in Analysis.* Dover. A reprint of the third North Holland edition, 1970 (first edition, 1958).

Dieudonné, J. (1968). *Calcul Infinitésimal.* Hermann, Paris. English translation: Houghton Mifflin, Boston (1971).

Écalle, J. (1992). *Introduction aux fonctions analysables et preuve constructive de la conjecture de Dulac.* Actualités mathématiques. Hermann, Paris.

Flajolet, P., Salvy, B., Zimmermann, P. (1991). Automatic average-case analysis of algorithms. *Theoretical Computer Science, Series A*, 79(1):37–109.

Geddes, K. O., Gonnet, G. H. (1989). A new algorithm for computing symbolic limits using hierarchical series. In Gianni, P., editor, *Symbolic and Algebraic Computation*, volume 358 of *Lecture Notes in Computer Science*, pages 490–495, New York. Springer-Verlag. Proceedings ISSAC'88, Rome.

Gruntz, D. (1996). *On Computing Limits in a Symbolic Manipulation System.* PhD thesis, ETH, Zürich.

Hardy, G. H. (1910). *Orders of Infinity*, volume 12 of *Cambridge Tracts in Mathematics.* Cambridge University Press.

Hardy, G. H. (1911). Properties of logarithmico-exponential functions. *Proceedings of the London Mathematical Society*, 10(2):54–90.

Harper, L. H. (1967). Stirling behavior is asymptotically normal. *Annals of Mathematical Statistics*, 38:410–414.

Harris, B., Schoenfeld, L. (1968). Asymptotic expansions for the coefficients of analytic functions. *Illinois Journal of Mathematics*, 12:264–277.

Hayman, W. K. (1956). A generalization of Stirling's formula. *Journal für die reine und angewandte Mathematik*, 196:67–95.

Meyer, R. E. (1980). Exponential asymptotics. *SIAM Review*, 22(2):213–224.

Odlyzko, A. M., Richmond, L. B. (1985). Asymptotic expansions for the coefficients of analytic generating functions. *Aequationes Mathematicae*, 28:50–63.

Péladan-Germa, A. (1995). Testing identities of series defined by algebraic partial differential equations. In Cohen, G., Giusti, M., Mora, T., editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, number 948 in Lecture Notes in Computer Science, pages 393–407. Springer-Verlag. Proceedings of the 11th International Symposium, AAECC-11, Paris, France, July 1995.

Richardson, D., Salvy, B., Shackell, J., Van der Hoeven, J. (1996). Asymptotic expansions of exp-log functions. In Lakshman, Y. N., editor, *Symbolic and Algebraic Computation*, pages 309–313, New York. ACM Press. Proceedings ISSAC'96, Zürich, July 1996.

Robinson, A. (1972). On the real closure of a Hardy field. In Asser, G., Flachsmeyer, J., Rinow, W., editors, *Theory of sets and topology*, pages 427–433, Berlin. VEB Deutsch. Verlag Wissensch.

Rosenlicht, M. (1983a). Hardy fields. *Journal of Mathematical Analysis and Applications*, 93(2):297–311.

Rosenlicht, M. (1983b). The rank of a Hardy field. *Transactions of the American Mathematical Society*, 280(2):659–671.

Rosenlicht, M. (1984). Rank change on adjoining real powers to Hardy fields. *Transactions of the American Mathematical Society*, 284(2):829–836.

Rosenlicht, M. (1987). Growth properties of functions in Hardy fields. *Transactions of the American Mathematical Society*, 299(1):261–272.

Rothstein, M., Caviness, B. F. (1979). A structure theorem for exponential and primitive functions. *SIAM Journal on Computing*, 8(3):357–367.

Sachkov, V. N. (1995). *Probabilistic Methods in Discrete Mathematics*, volume 56 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press.

Salvy, B. (1991a). *Asymptotique automatique et fonctions génératrices*. Ph. D. thesis, École polytechnique.

Salvy, B. (1991b). Examples of automatic asymptotic expansions. *SIGSAM Bulletin*, 25(2):4–17.

Salvy, B. (1994). Fast computation of some asymptotic functional inverses. *Journal of Symbolic Computation*, 17:227–236.

Salvy, B., Shackell, J. (1992). Asymptotic expansions of functional inverses. In Wang, P. S., editor, *Symbolic and Algebraic Computation*, pages 130–137, New York. ACM Press. Proceedings of ISSAC'92, Berkeley, July 1992.

Shackell, J. (1989). A differential-equations approach to functional equivalence. In Gonnet, G., editor, *Symbolic and Algebraic Computation*, pages 7–10, New York. ACM Press. Proceedings ISSAC'89, Portland.

Shackell, J. (1990). Growth estimates for exp-log functions. *Journal of Symbolic Computation*, 10(6):611–632.

Shackell, J. (1993a). Inverses of Hardy L-functions. *Bulletin of the London Mathematical Society*, 25:150–156.

Shackell, J. (1993b). Rosenlicht fields. *Transactions of the American Mathematical Society*, 335(2):579–595.

Shackell, J. (1993c). Zero-equivalence in function fields defined by algebraic differential equations. *Transactions of the American Mathematical Society*, 336(1):151–171.

Shackell, J. (1995). Extensions of asymptotic fields via meromorphic functions. *Journal of the London Mathematical Society*, 52:356–374.

Shackell, J. (1996). Limits of Liouvillian functions. *Proceedings of the London Mathematical Society*, 72(1):124–156.

Van den Dries, L., Macintyre, A., Marker, D. (1997). Logarithmic-exponential power series. *Journal of the London Mathematical Society*. To appear.

Van der Hoeven, J. (1994). General algorithms in asymptotics I: Gonnet and Gruntz's algorithm & II: Common operations. Technical Report LIX/RR/94/10, École polytechnique, Palaiseau, France.

Van der Hoeven, J. (1997). *Asymptotique Automatique*. PhD thesis, École polytechnique, Palaiseau, France.

Wilkie, A. J. (1996). Model completeness results for expansions of the ordered field of real numbers by restricted Pfaffian functions and the exponential function. *Journal of the American Mathematical Society*, 9(4):1051–1094.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105,
78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS
Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
(France)
http://www.inria.fr
ISSN 0249-6399