THE SIGSAM CHALLENGES: SYMBOLIC ASYMPTOTICS IN PRACTICE

PHILIPPE FLAJOLET AND BRUNO SALVY

Introduction

We present answers to 5 out of 10 of the "Problems, Puzzles, Challenges" proposed by G. J. Fee and M. B. Monagan in the March 1997 issue of the *Sigsam Bulletin*. In all cases, the answer to a seemingly numerical problem is obtained via series expansions and asymptotic methods. This illustrates more generally the crucial rôle played in the presence of singular behaviours by symbolic asymptotics as a bridge between symbolic computation and numerical computations.

All our computations have been performed using MapleV.4 on a Dec Alpha (255/233). The timings indicated correspond to executions on this machine.

Problem 2

What is the value of

$$\int_1^6 x^{x^x}\,dx$$

to 7 significant digits?

The integrand has a very fast increase, which makes it a good candidate for an asymptotic expansion using integration by parts (see [2, 8] for a more general treatment). We first illustrate the idea on a generic example:

```
F:=Int(exp(f(x)),x=a..A);
```

$$F := \int_a^A e^{f(x)}\,dx$$

It is assumed that $f(x)$ has fast increase. Then the integral is concentrated in the neighbourhood of the upper limit of integration. Now, we integrate by parts, rewriting the integrand as $f'e^f/f' = (e^f)'/f'$.

```
student[intparts](F,1/diff(f(x),x));
```

$$\frac{e^{f(A)}}{\frac{\partial}{\partial A}f(A)} - \frac{e^{f(a)}}{\frac{\partial}{\partial a}f(a)} + \int_a^A \frac{(\frac{\partial^2}{\partial x^2}f(x))e^{f(x)}}{(\frac{\partial}{\partial x}f(x))^2}\,dx$$

With $e^{f(x)} = x^{x^x}$, it is not difficult to see that the new integrand is increasing, and therefore the integral is bounded by $(A-a)$ times the value of the integrand at A, which is smaller than the first term by a factor of $f''(a)/f'(a)$. Iterating the same process, we get:

```
student[intparts](",diff(f(x),x,x)/diff(f(x),x)^3);
```

Date: Summer 1997.

$$\frac{e^{f(A)}}{\frac{\partial}{\partial A}f(A)} - \frac{e^{f(a)}}{\frac{\partial}{\partial a}f(a)} + \frac{(\frac{\partial^2}{\partial A^2}f(A))e^{f(A)}}{(\frac{\partial}{\partial A}f(A))^3} - \frac{(\frac{\partial^2}{\partial a^2}f(a))e^{f(a)}}{(\frac{\partial}{\partial a}f(a))^3}$$
$$- \int_a^A \left(\frac{\frac{\partial^3}{\partial x^3}f(x)}{(\frac{\partial}{\partial x}f(x))^3} - 3\frac{(\frac{\partial^2}{\partial x^2}f(x))^2}{(\frac{\partial}{\partial x}f(x))^4} \right) e^{f(x)}\,dx$$

Again, the new integrand has fast increase and is bounded by its value at the upper limit, which is smaller than the third term of this sum by a factor which can be computed. The process can again be iterated to get more and more accuracy. In the specific case of $e^{f(x)} = x^{x^x}$, $A = 6$, $a = 1$, the terms in $f(a)$ are so small that they can be neglected, and the computation proceeds as follows:

```
st:=time():
f:=x^(x^x); g:=normal(1/diff(log(f),x));
```

$$f := x^{x^x}$$
$$g := \frac{x}{x^x(\ln(x)^2x + \ln(x)x + 1)}$$

```
G:=Int(f,x);
```

$$G := \int x^{x^x}\,dx$$

```
for i to 3 do
    part:=normal(op(1,(-1)^(i+1)*G)*g/f);
    F:=student[intparts](G,part);
    G:=(-1)^i*op(select(type,indets(F,function),
        specfunc(anything,Int)));
    res[i]:=F-G;
od:
res:=[seq(res[i],i=1..3)];
```

$$res := [\frac{xx^{x^x}}{x^x(\ln(x)^2x + \ln(x)x + 1)},$$
$$\frac{(3\ln(x)x - 1 + \ln(x)^3x^2 + 2\ln(x)^2x^2 + \ln(x)x^2 + 2x)xx^{x^x}}{(x^x)^2(\ln(x)^2x + \ln(x)x + 1)^3},$$
$$(-2\ln(x)^2x - 16\ln(x)x - 2\ln(x)^3x^2 + 13\ln(x)^2x^2 + 21\ln(x)x^2$$
$$+ 9x^2 + 11\ln(x)^4x^3 + 30\ln(x)^3x^3 + 27\ln(x)^2x^3 + 8\ln(x)x^3$$
$$+ 12\ln(x)^4x^4 + 8\ln(x)^3x^4 + 8\ln(x)^5x^4 + 2\ln(x)^2x^4$$
$$+ 2\ln(x)^6x^4 + 1 - 12x)xx^{x^x} \Big/ ((x^x)^3\ln(x)^2x + \ln(x)x + 1)^5]$$

The numerical values given by these three terms are easy to compute:

```
evalf(subs(x=6.,res),10);
```

$$[.1102651583\,10^{36301}, \quad .1341606998\,10^{36296}, \quad .3239231242\,10^{36291}]$$

hence their sum is

```
evalf(convert(",`+`),7);
```

$$.1102665\,10^{36301}$$

The error is bounded by the value of the remaining integral, itself bounded by $(A-a)$ times its value at A:

```
bound_remainder:=evalf((6-1)*subs(x=6.,-op(1,-G)));
```

$$bound\_remainder := .1410589709\,10^{36293}$$

It follows that all the digits of the result obtained above are correct.

The total time required by this computation is:

```
time()-st;
```

$$.975$$

Thus the result is $.1102665\,10^{36301}$ up to 7 significant digits. The computation requires 3Mb of memory and less than 1 sec.

This example is typical of the use of asymptotic expansions in computer algebra. In practical computations, a closed-form seldom exists. Those problems which are challenging for a direct numerical computation can often be attacked by symbolic asymptotics. In examples like this one, the automation of the process requires a good handling of asymptotic scales. More and more is known in this direction (see e.g., [6]).

Problem 4

What is the coefficient of x^{3000} in the expansion of the polynomial

$$(x+1)^{2000}(x^2+x+1)^{1000}(x^4+x^3+x^2+x+1)^{500}$$

. to 13 significant digits?

This polynomial p is of degree 6000 and the sum of all its coefficients (do $x = 1$ in p) is a number of 1428 digits. Thus, on most sytems, this will induce large memory requirements, generate heavy computation, and perhaps even entail failure to return a result.

This problem may seem artificial; however, such questions turn up systematically in the reversion of power series since the inversion theorem of Lagrange states that the nth coefficient in the expansion of the inverse of a function f is expressible as an nth coefficient in an expression that involves an nth power of f.

We use the **gfun** package [7][1] that addresses the problem of manipulating series that satisfy linear differential equations with polynomial or rational coefficients (these are often called "holonomic" functions). From the point of view of symbolic manipulation, the importance of this package lies in the fact that: (i) a great many special functions of analysis (rational, algebraic, trigonometric, hypergeometric, etc) lie in the holonomic class; (ii) the class enjoys rich closure properties; (iii) identities are decidable and many fast numerical algorithms apply.

We attack the challenge by first computing, with **gfun**, a linear differential equation satisfied by p. We then get a linear recurrence equation satisfied by the coefficients of p, so that the coefficient of x^n in p becomes computable in a number of arithmetic operations that is linear in the quantity n. Heavy use is made throughout of the implementation of closure operations that are available via **gfun**.

First, we load the **gfun** package.

```
with(gfun):
```

Next, we specify each of the three factors of p by a (trivial) differential equation that it satisfies.

---

[1] The latest version is available at the URL http://www-rocq.inria.fr/algo/libraries

```
st:=time():
b1:={diff(y1(z),z)*(1+z)-2000*y1(z)=0,y1(0)=1}:
b2:={diff(y2(z),z)*(1+z+z^2)-1000*(1+2*z)*y2(z)=0,y2(0)=1}:
b3:={diff(y3(z),z)*(1+z+z^2+z^3+z^4)
    -500*(1+2*z+3*z^2+4*z^3)*y3(z)=0,y3(0)=1}:
```

Next, we apply closure operations, here `gfun[poltodiffeq]`:

```
b123:=poltodiffeq(y1(z)*y2(z)*y3(z),[b1,b2,b3],
        [y1(z),y2(z),y3(z)],Y(z));
```

$$b123 := \{(-3500-10000z-16500z^2-19500z^3-20000z^4-14500z^5-6000z^6)Y(z) + (1+3z+5z^2+6z^3+6z^4+5z^5+3z^6+z^7)\frac{\partial}{\partial z}Y(z), Y(0)=1\}$$

This gives rise to a simple recurrence on coefficients, by `gfun[diffeqtorec]`:

```
r123:=diffeqtorec(b123,Y(z),u(n));
```

$$r123 := \{u(1)=3500, u(2)=6124750, u(3)=7144958500, u(4)=6251073531125, u(5)=4375037588062700, u(6)=2551584931812376500, u(0)=1, (n-6000)u(n)+(3n-14497)u(n+1)+(5n-19990)u(n+2) + (6n-19482)u(n+3)+(6n-16476)u(n+4)+(5n-9975)u(n+5) + (3n-3482)u(n+6)+(n+7)u(n+7)\}$$

This is then converted to a procedure by `gfun[rectoproc]`:

```
ci:=rectoproc(r123,u(n));
```

$ci :=$ **proc**(n)
  local $i, u0, u1, u2, u3, u4, u5, u6, u7$;
    $u0 := 1$;
    $u1 := 3500$;
    $u2 := 6124750$;
    $u3 := 7144958500$;
    $u4 := 6251073531125$;
    $u5 := 4375037588062700$;
    $u6 := 2551584931812376500$;
    for i **from** 7 **to** $n-1$ **do**
      $u7 := -(-6007 \times u0 - 14518 \times u1 - 20025 \times u2 - 19524 \times u3 - 16518 \times u4 - 10010 \times u5 - 3503 \times u6 + (u0 + 3 \times u1 + 5 \times u2 + 6 \times u3 + 6 \times u4 + 5 \times u5 + 3 \times u6) \times i)/i$;
      $u0 := u1; u1 := u2; u2 := u3; u3 := u4; u4 := u5; u5 := u6; u6 := u7$
    od;
    $-(-6007 \times u0 - 14518 \times u1 - 20025 \times u2 - 19524 \times u3 - 16518 \times u4 - 10010 \times u5 - 3503 \times u6 + (u0 + 3 \times u1 + 5 \times u2 + 6 \times u3 + 6 \times u4 + 5 \times u5 + 3 \times u6) \times n)/n$
  end

```
time()-st;
```

$$1.605$$

The whole preprocessing stage has taken 1.6 seconds. We can now use this procedure to compute coefficients efficiently.

```
st:=time():ci3000:=ci(3000);TIME=time()-st;
```

ci3000 := 397394226558004303969667626329928604465422787974692638585\
186649500492257098807566041611920517888566124389324550428942422\61\
985201809225844217500721836490767140642033100836325677674151720\67\
841999633718492797074653404851457323010106410500986842742573651\99\
137822372444333612694235035865919234133652674977465226120423435\79\
985520798460086148278667331447447900625876453816149920656399289\89\
221762135281216412663929705475090445207788633306766757562601722\20\
682527359303517400527070880915279789742055429074734240165217337\09\
675132804414472786234412637595065021742410628691722061426399531\66\
543897796174957722171626492417218827927494514181584672198036085\74\
787209256948729083871163148476352370777889846254639890233124984\35\
185579132822835658957110647517645591748028391905915219208230225\18\
871748226066088801001499437002574934771671473841782258918898203\53\
855858121209322359972399432954464940680910666400538389190479567\69\
726600774734181120871309444752033390108283206475790871301280547\70\
918373778023467597005690374647836154867429544835365734889910710\68\
715494214070314614623037197913047820183426928328820435915100231\10\
312930052425314708952930677843324950645160145890055548997481975\63\
492058353540645071064052314677953555756703028203890234165386314\71\
813786584080886901650502270553674168316961093490327289351780477\34\
770817123168416066379290841435679293732047395153722336538489757\92\
803112770485725600338933543494692723377803871679071342944579342\04\
76320

$$TIME = 13.147$$

The computation requires only 13 seconds of CPU time. The value of the result is approximately $3.97\ 10^{1427}$.

Note that, by starting the computation with floating point numbers, the whole computation is performed using the accuracy given by the Maple environment variable `Digits`. Here, the recurrence is stable and the computation time drops to 3.6 seconds.

```
st:=time():Digits:=14:cf:=subs(3500=3500.0,op(ci)):
cf3000:=cf(3000);TIME=time()-st;
```

$$cf3000 := .39739422655366\ 10^{1427}$$

$$TIME = 3.590$$

```
(cf3000-ci3000)/ci3000;
```

$$-.10921145074478\,10^{-10}$$

Such techniques are used for instance for estimating the probability of occurrence of patterns in DNA sequences, under a simple model of randomness. In this case, the generating function is rational and it is required to find coefficients of large orders (often several thousands) that are dictated by the lengths of DNA fragments under consideration[2].

Problem 5

What is the largest zero of the 1000th Laguerre polynomial to 12 significant digits?

The asymptotics of the zeroes of the Laguerre polynomials are known and could be used to compute the result efficiently. It is also possible to use a computer algebra system to perform the computations leading to the corresponding asymptotic expansion. We refer to Abramowitz & Stegun [1] and to Szegö [9] for details on this approach.

Instead, we proceed numerically using two informations:

- the explicit form of the nth Laguerre polynomial:

$$L_n(x) = \sum_{k=0}^{n} \binom{n}{k} \frac{(-1)^k x^k}{k!};$$

- the well-known fact that the roots are positive real numbers.

A classical technique to compute the largest root of a polynomial is the Graeffe process [3]. It is based on the iteration $P(x) \mapsto P(\sqrt{x})P(-\sqrt{x})$, which transforms a polynomial P into a polynomial of the same degree whose roots are the squares of the roots of P. Thus the square root of the sum of the roots of P is closer to the largest root of P than the sum of the roots of P. Iterating this process converges to the largest root of P when, as in the Laguerre polynomials, there is only one root of largest modulus. When all the roots of P are positive real numbers, it is easy to see that each of the approximations provided by this method is larger than the actual root.

For polynomials of large degree, the mere computation of a product is very time-consuming. In this particular example, Maple needs 44710 sec. to compute the product used in the first iteration of the Graeffe process. However, since we are only interested in the sum of the roots of these iterated polynomials, not all the coefficients of these products need be computed. For instance, to compute both leading coefficients of the product of two polynomials, it is only necessary to know two coefficients of each of them. Thus to compute eight iterations of the Graeffe process, we only need $2^8 = 256$ coefficients of L_{1000}.

To simplify the computation, we work with the reciprocal of the polynomials:

```
n:=1000:
nb_iter:=8:
nb_terms:=2^nb_iter:
c[0]:=1: for i to nb_terms do c[i]:=-(n-i+1)^2/i*c[i-1] od:
```

This is the series expansion of $x^{1000}L_{1000}(x^{-1})$ to the order 256:

```
S:=add(c[i]*x^i,i=0..nb_terms):
```

[2]See the URL http://www-rocq.inria.fr/algo/libraries/autocomb.

We now compute the Graeffe iterations and print the corresponding estimate for the largest root:

```
for i to nb_iter do
    S:=series(S*subs(x=-x,S),x,nb_terms+1);
    nb_terms:=iquo(nb_terms,2);
    for j from 0 to nb_terms do c[j]:=coeff(S,x,2*j) od;
    S:=add(c[j]*x^j,j=0..nb_terms);
    estimate:=evalf((-coeff(S,x,1)/coeff(S,x,0))^(1/2^i));
    print(i,estimate)
od:
```

$$
\begin{array}{ll}
1, & 44710.17781 \\
2, & 10871.93993 \\
3, & 5875.424153 \\
4, & 4557.080810 \\
5, & 4131.575859 \\
6, & 3992.655417 \\
7, & 3952.663782 \\
8, & 3944.199951
\end{array}
$$

As mentioned above, this sequence of values provides increasingly good estimates for the largest root, which is smaller than all of them. At this point, it is a good idea to use the Newton-Raphson method with the last estimate as initial value. The method will converge to the largest root since the polynomial has positive first and second derivatives in that area by Rolle's theorem.

The following procedure evaluates $L_{1000}(x)$ and its derivative. Also, it returns the largest summand encountered during the computation, which makes it possible to check *a posteriori* whether the precision used in the intermediate computations was sufficient.

```
lag1000 := proc(x)
    local c, k, d0, d1, maxc;
        c := 1; d0 := 1; d1 := 0; maxc := 1;
        for k to 1000 do
            c := (k − 1001) × c/k;
            if maxc < (−1)^k × c then maxc := (−1)^k × c fi;
            d1 := d1 + c;
            c := c × x/k;
            d0 := d0 + c
        od;
        [d0, d1, maxc]
    end
```

Experiments show that a precision of 300 digits is sufficient.

```
Digits:=300:
```

We then just have to run the Newton iteration a few times:

```
for i to 7 do
    r:=lag1000(estimate);
```

```
    estimate:=estimate-r[1]/r[2];
    print(i,time(),evalf(estimate,40))
od:
```

1,	15.007,	3943.554169639569910236033441208347213568
2,	16.913,	3943.288156587100771812724414055574813853
3,	18.903,	3943.248208728579495923271549680060251158
4,	20.960,	3943.247395176254653907520502075927102493
5,	23.011,	3943.247394845271007150933475723163321646
6,	25.103,	3943.247394845270952389728107754944655528
7,	27.212,	3943.247394845270952389728107753445640963

```
evalf(estimate,12);
```

3943.24739485

It takes five iterations and a total of 23 sec. since the beginning of the session to get 12 significant digits. Each new iteration then takes approximately 2.2 sec. and roughly doubles the number of significant digits. A lot more precision can then be attained by increasing Digits for each iteration. The computation requires approximately 55 Mb of memory. This huge memory consumption comes from the product of series which is not as efficient as one would like it in Maple.

Problem 7

Define functions f and g as follows:

$$\begin{aligned} f(x) = {} & \tan(\tanh(\sin(x))) - \tan(\sin(\tanh(x))) \\ & + \tan(\tanh(\sinh(x))) - \tan(\sinh(\tanh(x))) \\ & + \tanh(\sin(\tan(x))) - \tanh(\tan(\sin(x))) \\ & + \sin(\tan(\tanh(x))) - \sin(\tanh(\tan(x))) \\ & + \sinh(\tan(\tanh(x))) - \sinh(\tanh(\tan(x))) \\ & + \tanh(\sinh(\tan(x))) - \tanh(\tan(\sinh(x))) \\ g(x) = {} & \sinh(\tanh(\sin(x))) - \sinh(\sin(\tanh(x))) \\ & + \tanh(\sin(\sinh(x))) - \tanh(\sinh(\sin(x))) \\ & + \sin(\sinh(\tanh(x))) - \sin(\tanh(\sinh(x))) \\ & + \tan(\sin(\sinh(x))) - \tan(\sinh(\sin(x))) \\ & + \sin(\sinh(\tan(x))) - \sin(\tan(\sinh(x))) \\ & + \sinh(\tan(\sin(x))) - \sinh(\sin(\tan(x))) \end{aligned}$$

What is

$$\lim_{x\to 0} \frac{f(g(x))}{g(f(x))}$$

to 9 significant digits?

This is simply obtained by series expansions to a moderate order:

```
st:=time():
F:=tan(tanh(sin(x)))-tan(sin(tanh(x)))+
    tan(tanh(sinh(x)))-tan(sinh(tanh(x)))+
```

```
    tanh(sin(tan(x)))-tanh(tan(sin(x)))+
    sin(tan(tanh(x)))-sin(tanh(tan(x)))+
    sinh(tan(tanh(x)))-sinh(tanh(tan(x)))+
    tanh(sinh(tan(x)))-tanh(tan(sinh(x))):
num:=series(F,x,17);
```

$$num := \frac{7769}{3274425}x^{15} + O(x^{17})$$

```
G:=sinh(tanh(sin(x)))-sinh(sin(tanh(x)))+
    tanh(sin(sinh(x)))-tanh(sinh(sin(x)))+
    sin(sinh(tanh(x)))-sin(tanh(sinh(x)))+
    tan(sin(sinh(x)))-tan(sinh(sin(x)))+
    sin(sinh(tan(x)))-sin(tan(sinh(x)))+
    sinh(tan(sin(x)))-sinh(sin(tan(x))):
den:=series(G,x,12);
```

$$den := \frac{4}{63}x^{11} + O(x^{12})$$

```
series(subs(x=den,num)/subs(x=num,den),x,infinity);
```

$$\frac{24514478609520577408170967296000000000000000000000}{80103448751723203083149895150908444280 1} + O(x)$$

```
evalf(op(1,"),10);
```

$$.3060352456\ 10^{10}$$

```
time()-st;
```

$$1.301$$

This computation requires less than 2Mb and 1.3sec.

It would be desirable to have the system compute automatically the first order expansion, increasing the order of intermediate expansions if necessary. The difficulty there consists in recognizing that the function under consideration is different from 0. In sufficient generality this is undecidable [4], but these functions f and g fall into a class where this can be done modulo a zero-equivalence test for elementary constants [5].

PROBLEM 8

What is

$$\prod_{n=1}^{\infty} \tanh(\frac{1}{2}\arctan(n)\sinh^{-1}(n))$$

to 14 significant digits?

As the analysis below shows, this product is convergent. Its terms tend to 1 reasonably fast, so that the numerical value can be obtained from the computation of a truncation of the product and an asymptotic analysis of the rest. By taking a logarithm, we can work with a sum rather than a product. Then the general principle is as follows [10, p. 163]: if $f(z) = \sum_{n>1} f_n z^n$, then formally at least

$$\sum_{k=1}^{\infty} f(1/k) = \sum_{n=2}^{\infty} f_n \zeta(n).$$

We thus start by computing an asymptotic expansion of the log of the nth term. This is made slightly difficult by the inability of the current version of Maple to handle non-rational exponents in the expansions.

```
st:=time():
Order:=20:
S:=asympt(1/2*arctan(n)*arcsinh(n),n);
```

$$S := \frac{1}{4}\pi(\ln(2)+\ln(n)) + \frac{-\frac{1}{2}\ln(2)-\frac{1}{2}\ln(n)}{n} + \frac{1}{16}\frac{\pi}{n^2} + \frac{-\frac{1}{8}+\frac{1}{6}\ln(2)+\frac{1}{6}\ln(n)}{n^3} - \frac{3}{128}\frac{\pi}{n^4} + \cdots$$

and more terms that we do not display which follow the same pattern.

We first isolate the leading term which we shall treat specially.

```
S1:=Pi/4*ln(n);
```

$$S1 := \frac{1}{4}\pi\ln(n)$$

```
S2:=S-S1:
convert(tanh(x),exp);
```

$$\frac{(e^x)^2-1}{(e^x)^2+1}$$

We use $n^{3/4}/T(n)$ to denote $e^{S_1} = n^{\pi/4}$, which implies that $T(n)$ is of asymptotic order $n^{-.035}$. Besides, we set $n = N^2$ in order to work with polynomials later. We start by computing the expansion of e^{-2S}:

```
uu:=asympt(exp(-2*subs(n=N^2,S2))*T^2/N^3,N):
```

Then we substitute this in the expansion of $\ln(\tanh(x))$:

```
res:=asympt(subs(x=uu,ln((1-x)/(1+x))),N):
```

We are now in a position to sum these terms for n from M to infinity, for some M that we'll choose later. There we use the property that

$$\sum_{n=M}^{\infty} \frac{(-1)^k \log^k(n)}{n^s} = \zeta^{(k)}(s) - \sum_{n=1}^{M-1} \frac{(-1)^k \log^k(n)}{n^s},$$

of which Maple's **sum** function is only aware when $k = 0$. We change the variable to a variable tending to 0 in order to work with polynomial coefficients. The notation $\bar{\zeta}$ below denotes the truncated sum on the left.

```
res:=subs(ln(N)=-U/2,N=1/nn,convert(res,polynom)):
for i from 3 by 2 to Order do
    p:=expand(coeff(res,nn,i));
    lco:=[coeffs(p,[T,U],'lexp')];lexp:=[lexp];
    Res[i]:=add(lco[j]*Zetabar(degree(lexp[j],U),
        degree(lexp[j],T)*(Pi/4-3/4)+i/2),j=1..nops(lco))
od:
Res:=[seq(Res[1+2*i],i=1..iquo(Order,2)-1)];
```

$$Res := [-2\frac{\bar\zeta(0,\pi/2)}{\sqrt{2^\pi}}, 2\frac{\bar\zeta(1,1+\pi/2)}{\sqrt{2^\pi}} - 2\frac{\ln(2)\bar\zeta(0,1+\pi/2)}{\sqrt{2^\pi}},$$
$$-\frac{\bar\zeta(2,2+\pi/2)}{\sqrt{2^\pi}} + 2\frac{\ln(2)\bar\zeta(1,2+\pi/2)}{\sqrt{2^\pi}} + (\frac{1}{4}\frac{\pi}{\sqrt{2^\pi}} - \frac{\ln(2)^2}{\sqrt{2^\pi}})\bar\zeta(0,2+\pi/2),$$
$$\ldots$$

where the dots indicate several pages of the same kind of output.

The terms of this list are the successive contributions of the asymptotic expansion. We now evaluate them for $M = 100$ and then multiply by the first terms of the product to obtain the desired approximation.

```
`evalf/Zetabar` := proc(k, s)
    local i, M;
    global Log, ims;
        M := 100;
        if not assigned(ims[M − 1, s]) then
            for i to M − 1 do ims[i, s] := evalf(i^(−s)) od fi;
        if k ≠ 0 then
            if not assigned(Log[1, M − 1]) then
            for i to M − 1 do Log[1, i] := evalf(log(i)) od fi;
            if not assigned(Log[k, M − 1]) then
            for i to M − 1 do Log[k, i] := Log[1, i]^k od fi;
            evalf(ζ(k, s)) − (−1)^k × add(Log[k, i] × ims[i, s], i = 1..M − 1)
        else evalf(ζ(s)) − add(ims[i, s], i = 1..M − 1)
        fi
    end
```

The symbolic part of the computation has taken the following time:

```
time()-st;
```

$$1.811$$

We now turn to the numerical computation:

```
Digits:=30:
res:=evalf(Res);
```

$$\begin{aligned} res := [&-.085376883881769421347726759280 2,\\ &-.0018489416992978086972397828763 0,\\ &-.000030303964455894516119507013079 7,\\ &-.350526064332901053341251097304\,10^{-6},\\ &-.268080484811869025819048283146\,10^{-8},\\ &-.110204653888531983348874288239\,10^{-10},\\ &-.470151160346022439444595984011\,10^{-13},\\ &-.271482094920976202629908971233\,10^{-14},\\ &-.909829054007018877303913867423\,10^{-16}]\end{aligned}$$

These are the successive terms of the approximation of the truncated sum. Here is the end result:

```
exp('+'(op(res)))*mul(evalf(tanh(arctan(i)*arcsinh(i)/2)),i=1..99):
evalf(",14);
```

$$.084439684030189$$

The total time required for this computation is

```
time()-st;
```

$$30.381$$

The same method applies to Problem 3, but is rendered more delicate by the non-rational exponents that have to be dealt with, like $\pi/4$ in the present example.

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

References

[1] Abramowitz, M., and Stegun, I. A., Eds. *Handbook of mathematical functions with formulas, graphs, and mathematical tables.* Dover Publications Inc., New York, 1992. Reprint of the 1972 edition.

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

[3] Henrici, P. *Applied and computational complex analysis, Vol. 1.* Wiley Classics Library. John Wiley & Sons Inc., New York, 1988. Reprint of the 1974 original.

[4] Richardson, D. Some undecidable problems involving elementary functions of a real variable. *Journal of Symbolic Logic 33*, 4 (1968), 514–520.

[5] Richardson, D. The elementary constant problem. In *Symbolic and Algebraic Computation* (New York, 1992), P. S. Wang, Ed., ACM Press. Proceedings of ISSAC'92, Berkeley, July 1992.

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

[7] Salvy, B., and Zimmermann, P. Gfun: a Maple package for the manipulation of generating and holonomic functions in one variable. *ACM Transactions on Mathematical Software 20*, 2 (1994), 163–177.

[8] Sirovich, L. *Techniques of Asymptotic Analysis*, vol. 2 of *Applied Mathematical Sciences.* Springer-Verlag, New York, 1971.

[9] Szegö, G. *Orthogonal Polynomials*, fourth ed., vol. XXIII of *Colloquium Publications.* American Mathematical Society, Providence, R.I., 1975.

[10] Vardi, I. *Computational Recreations in Mathematica.* Addison-Wesley Publishing Company Advanced Book Program, Redwood City, CA, 1991.

Algorithms Project, INRIA Rocquencourt, F-78153 Le Chesnay, France
E-mail address: Philippe.Flajolet@inria.fr, Bruno.Salvy@inria.fr