

# Numerical methods and arbitrary-precision computation of the Lerch transcendent

Guillermo Navas-Palencia

## Abstract

We examine the use of the Euler-Maclaurin formula and new derived uniform asymptotic expansions for the numerical evaluation of the Lerch transcendent  $\Phi(z, s, a)$  for  $z, s, a \in \mathbb{C}$  to arbitrary precision. A detailed analysis of these expansions is accompanied by rigorous error bounds. A complete scheme of computation for large and small values of the parameters and argument is described along with algorithmic details to achieve high performance. The described algorithm has been extensively tested in different regimes of the parameters and compared with current state-of-the-art codes. An open source implementation of  $\Phi(z, s, a)$  based on the algorithms described in this paper is available.

## 1 Introduction

The Lerch transcendent, also called Hurwitz-Lerch zeta function, which is named after the Czech mathematician Mathias Lerch (1860 - 1922) is defined by means of the Dirichlet series [1]

$$\Phi(z, s, a) = \sum_{k=0}^{\infty} \frac{z^k}{(k+a)^s}, \quad (1)$$

where  $\Phi(z, s, a)$  is absolutely convergent for  $|z| \leq 1$ ,  $a \notin \mathbb{Z}_0^-$  or  $\Re(s) > 1$ ,  $|z| = 1$  and is defined elsewhere by analytic continuation. The Lerch transcendent serves as a unified framework for the study of various particular cases of special functions in number theory such as polygamma functions, polylogarithms, Dirichlet  $L$ -functions and certain number-theoretical constants. The Lerch transcendent is related to Lipschitz-Lerch zeta function by the functional equation

$$\mathcal{R}(a, x, s) = \Phi(e^{2\pi ix}, s, a).$$

This function was introduced and investigated by Lerch [22] and Lipschitz [23], where the latter studied general Euler integrals including the Lerch zeta function. Subsequently, many authors have studied properties of these functions. Among the recent investigations on the analytic properties of Lerch zeta function, we remark the work conducted by Laurinćikas and Garunkštis in [21].

The Lerch transcendent and their special cases are ubiquitous in theoretical physics. They play a relevant role in particle physics, thermodynamics and statistical mechanics, being present, for instance, in Bose-Einstein condensation distribution [14] and integrals of the Fermi-Dirac distribution. They also occur in quantum field theory, in particular in quantum electrodynamic bound state calculations [17]. Regarding mathematical applications, the Lerch zeta function can be used to evaluate Dirichlet  $L$ -series of the form

$$L(s, \chi) = \sum_{k=1}^{\infty} \frac{\chi(k)}{k^s},$$

where  $\chi : (\mathbb{Z}/q\mathbb{Z})^* \rightarrow \mathbb{C}$  is a Dirichlet character and  $q$  a natural number, thus the above summation is also expressible as a combination of Hurwitz zeta functions  $\zeta(s, a)$  or polygamma functions for  $s \in \mathbb{N}$

$$L(s, \chi) = \sum_{r=1}^q \chi(r) \sum_{n=0}^{\infty} \frac{1}{(r+nq)^s} = q^{-s} \sum_{r=1}^q \chi(r) \zeta(s, r/q).$$

The Lerch transcendent occasionally occurs in statistics, for instance, it provides an analytic expression for the central moments of the geometric distribution.

Over the last two decades several authors have devised new series representations to extend the regime of computation of the Lerch transcendent. Complete asymptotic expansions including error bounds of  $\Phi(z, s, a)$  for large  $a$  and large  $z$  are derived in [9]. More recently, an *exponentially-improved* expansion for the Lerch zeta function in large  $a$  asymptotic was examined in [26]. A remarkable and extensive review of properties, identities and numerical methods for the computation of the Lerch transcendent and their special cases was carried out by R. Crandall in [5]. In addition, we mention two important convergent series: the Hasse's convergent series expansion in [15] given by

$$(1-z)\Phi(z, s, a) = \sum_{n=0}^{\infty} \left( \frac{-z}{1-z} \right)^n \sum_{k=0}^n (-1)^k \binom{n}{k} (a+k)^{-s},$$

which holds for  $s, z \in \mathbb{C}$  with  $\Re(z) < 1/2$  and Erdélyi-series representation [8]

$$z^a \Phi(z, s, a) = \sum_{k=0}^{\infty} \zeta(s-k, a) \frac{\log^k(z)}{k!} + \Gamma(1-s)(-\log(z))^{s-1},$$

where  $s$  is not a positive integer, and for parameter  $a \in (0, 1]$ ,  $|\log(z)| < 2\pi$ , the series representation is linearly convergent.

Finally, the Hermite-type integral representation is given by

$$\begin{aligned} \Phi(z, s, a) &= \frac{1}{2a^s} + \frac{(-\log(z))^{s-1}}{z^a} \Gamma(1-s, -a \log(z)) \\ &+ 2 \int_0^{\infty} \frac{\sin(s \arctan(t/a) - t \log(z))}{(a^2 + t^2)^{s/2} (e^{2\pi t} - 1)} dt, \quad \Re(a) > 0. \end{aligned} \quad (2)$$

In this paper, we derive complete new uniform asymptotic expansions of  $\Phi(z, s, a)$  for large order of the parameters  $a, s$  and argument  $z$ , with special emphasis on the less investigated case  $\Re(z) \gg 0$ . The starting point for our asymptotic expansions is the integral representation in (2). Additionally, a careful treatment of the Euler-Maclaurin formula is considered along with the calculation of a rigorous error bound. A significant effort have been made to develop uniform asymptotic expansions with tractable coefficients in terms of known entities and amenable to arbitrary-precision computations. An extensive discussion on algorithmic aspects for their successful implementation is also provided.

The outline of the paper is the following: in Section 2 we study the main numerical methods considered for the numerical evaluation of  $\Phi(z, s, a)$ , including error bounds. Then, in Section 3, we discuss in detail implementation aspects, several heuristics and performance issues. We also devise an effective algorithm that permits computation to arbitrary-precision in an extensive region of the function's domain. In Section 4, we provide numerical calculations and compare the present implementation with open source and commercial state-of-the-art libraries. Finally, in Section 5, we discuss possible enhancements and present our conclusions.

## 2 Numerical methods

### 2.1 Euler-Maclaurin formula

We briefly summarized the Euler-Maclaurin formula and refer to [2] for a formal proof. We closely follow the expository style in [19]. Let us suppose that  $f$  is an analytic function on a closed domain  $[N, U]$  where  $N, U \in \mathbb{Z}$ , and let  $M$  be a positive integer. Let  $B_n$  denote the  $n$ -th Bernoulli number and  $\tilde{B}_{2M}(t) = B_n(t - [t])$  denote the  $n$ -th periodic Bernoulli polynomials. The Euler-Maclaurin summation formula states that

$$\sum_{k=N}^U f(k) = I + T + R \quad (3)$$

where

$$I = \int_N^U f(t) dt \quad (4)$$

$$T = \frac{1}{2}(f(N) + f(U)) + \sum_{k=1}^M \frac{B_{2k}}{(2k)!} \left( f^{(2k-1)}(N) - f^{(2k-1)}(U) \right) \quad (5)$$

$$R = - \int_N^U \frac{\tilde{B}_{2M}(t)}{(2M)!} f^{(2M)}(t) dt. \quad (6)$$

If  $f$  decreases sufficiently rapid, letting  $U \rightarrow \infty$  the above equations remain valid.

**Proposition 2.1** *The Euler-Maclaurin summation formula for the Lerch transcendent is given by*

$$\Phi(z, s, a) = S + I + T + R, \quad (7)$$

where

$$S = \sum_{k=0}^{N-1} \frac{z^k}{(k+a)^s}, \quad (8)$$

$$I = \frac{(-\log(z))^{s-1}}{z^a} \Gamma(1-s, -(a+N)\log(z)), \quad (9)$$

$$T = \frac{z^N}{(a+N)^s} \left( \frac{1}{2} + \sum_{k=1}^M \frac{B_{2k}}{(2k)!} \frac{U(-2k+1, -2k+2-s, -(a+N)\log(z))}{(a+N)^{2k-1}} \right), \quad (10)$$

$$R = - \int_N^\infty \frac{\tilde{B}_{2M}(t)}{(2M)!} \frac{z^t}{(a+t)^{s+2M}} U(-2M, -2M+1-s, -(a+t)\log(z)) dt. \quad (11)$$

**Proof:** Let us first consider the Hermite-type integral in (2)

$$I := \int_0^\infty \frac{\sin(s \arctan(t/a) - t \log(z))}{(a^2 + t^2)^{s/2} (e^{2\pi t} - 1)} dt. \quad (12)$$

For  $z, s, a \in \mathbb{R}$ ,  $z > 0$  and  $a > 0$ , the above integral can be written in the form

$$I = \frac{1}{a^s} \Im \left( \int_0^\infty \frac{z^{-it}}{(1-it/a)^s e^{2\pi t} - 1} dt \right). \quad (13)$$

The domain delimited by previous constraints shall be extended by analytic continuation. Now we express the integrand in (13) in terms of the confluent hypergeometric function  $U(a, b, z)$  which yields

$$I = \frac{1}{a^s} \Im \left( (-a \log(z))^s \int_0^\infty \frac{e^{-i \log(t)} U(s, s+1, (it-a)\log(z))}{e^{2\pi t} - 1} dt \right), \quad (14)$$

By applying the addition theorem for  $U(a, b, z)$  [6, §13.13] given by

$$U(a, b, x + y) = e^y \sum_{n=0}^{\infty} \frac{(-y)^n}{n!} U(a, b + n, x), \quad |y| < |x|. \quad (15)$$

the integrand can be written as a summation defined by

$$e^{-i \log(z)} U(s, s + 1, (it - a) \log(z)) = \sum_{k=0}^{\infty} \frac{(-it \log(z))^k U(s, s + k + 1, -a \log(z))}{k!}. \quad (16)$$

Substituting (16) into (14) and formally interchanging summation and integration we obtain

$$I = \frac{1}{a^s} \Im \left( (-a \log(z))^s \sum_{k=0}^{\infty} \frac{(-i \log(z))^k U(s, s + k + 1, -a \log(z))}{k!} \int_0^{\infty} \frac{t^k}{e^{2\pi t} - 1} dt \right),$$

where the integral can be directly evaluated in closed form by

$$\int_0^{\infty} \frac{t^k}{e^{2\pi t} - 1} dt = \frac{k!}{(2\pi)^{k+1}} \zeta(k + 1).$$

We use Kummer's transformation  $U(s, s + k + 1, -a \log(z))(-a \log(z))^{k+s} = U(-k, 1 - k - s, -a \log(z))$  to rewrite  $I$  in the form

$$I = \frac{1}{a^s} \Im \left( \sum_{k=1}^{\infty} \frac{i^k}{a^k} \frac{U(-k, 1 - k - s, -a \log(z))}{(2\pi)^{k+1}} \zeta(k + 1) \right). \quad (17)$$

Note that the same summation formula can be derived by expanding  $f(t) = z^{-it}(1 - it/a)^{-s}$ , which gives

$$f(t) = \sum_{k=0}^{\infty} \left( \frac{-it}{a} \right)^k \frac{1}{k!} \sum_{j=0}^{\infty} \binom{k}{j} (-1)^j (a \log(z))^{k-j} \sum_{m=0}^j (-1)^{j-m} s(j, m) s^m, \quad (18)$$

where  $s(j, m)$  are Stirling numbers of the first kind. The inner summation in (18) is expressible in terms of rising factorial or Pochhammer's symbol  $(s)_j$  using the well-known identities

$$\sum_{m=0}^j (-1)^{j-m} s(j, m) s^m = (-1)^j (-s)^{(j)} = (s)_j \quad (19)$$

and

$$\sum_{j=0}^{\infty} \binom{k}{j} (a \log(z))^{k-j} (-s)^{(j)} = (-1)^k U(-k, 1 - k - s, -a \log(z)). \quad (20)$$

Finally, taking the imaginary part of (17) yields

$$\begin{aligned} I &= \frac{1}{a^s} \sum_{k=0}^{\infty} \frac{(-1)^k}{a^{2k+1}} \frac{U(-2k - 1, -2k - s, -a \log(z))}{(2\pi)^{2k+2}} \zeta(2k + 2) \\ &= \frac{1}{2a^s} \sum_{k=1}^{\infty} \frac{B_{2k}}{(2k)!} \frac{U(-2k + 1, -2k + 2 - s, -a \log(z))}{a^{2k-1}}, \end{aligned} \quad (21)$$

where the relationship between Bernoulli numbers  $B_{2k}$  and the Riemann zeta function is applied.  $\square$

Note that the expansion is convergent for  $|\log(z)| < 2\pi$ . This can be observed by taking the asymptotic estimate of the  $k$ -th term in (21)

$$|t_k| = \left| \frac{B_{2k}}{(2k)!} \frac{U(-2k+1, -2k+2-s, -(a+N)\log(z))}{(a+N)^{2k-1}} \right| \\ \sim \frac{2}{(2\pi)^{2k}} |\log(z)|^{2k-1}, \quad (22)$$

as  $N, M \rightarrow \infty$ , where we consider the usual asymptotic estimates for  $U(a, b, z) \sim z^{-a}$  as  $|z| \rightarrow \infty$  and  $|B_{2k}|/(2k)!$  using the fact that  $\zeta(2k) \sim 1$  as  $k \rightarrow \infty$ . To assess the domain of convergence for  $z$  we use the ratio test (d'Alembert ratio test)

$$\lim_{k \rightarrow \infty} \left| \frac{t_{k+1}}{t_k} \right| \sim \frac{|\log(z)|^2}{4\pi^2} < 1 \iff |\log(z)| < 2\pi.$$

Finally, taking  $M$  such that  $\Re(s) + 2M - 1 > 0$ , the remainder term (11) in the Euler-Maclaurin summation formula is well defined, giving its analytic continuation to  $s \in \mathbb{C} \setminus \{1\}$ .

**Theorem 2.2** *Given  $a, s, z \in \mathbb{C}$  with  $|\log(z)| < 2\pi$  and  $N, M \in \mathbb{N}$  such that  $\Re(a) + N > 0$  and  $\Re(s) + 2M > 1$ , the error term (11) in the Euler-Maclaurin summation formula satisfies*

$$|R| \leq \frac{4}{(2\pi)^{2M}} \left| C \sum_{k=0}^{2M} \binom{2M}{k} Q(k+1-2M-s, W) \frac{(-\log(z))^{-k-1+2M+s}}{\log^{-k}(z)} \right|, \quad (23)$$

where  $C = \Gamma(1-s)/z^a$ ,  $W = -(a+N)\log(z)$  and  $Q(a, z) = \Gamma(a, z)/\Gamma(a)$  is the regularized incomplete Gamma function.

**Proof:** We have

$$|R| = \left| \int_N^\infty \frac{\tilde{B}_{2M}(t)}{(2M)!} \frac{z^t}{(a+t)^{s+2M}} U(-2M, -2M+1-s, -(a+t)\log(z)) dt \right| \\ \leq \frac{|\tilde{B}_{2M}(t)|}{(2M)!} \left| \int_N^\infty \frac{z^t}{(a+t)^{s+2M}} U(-2M, -2M+1-s, -(a+t)\log(z)) dt \right| \\ \leq \frac{4}{(2\pi)^{2M}} \left| \sum_{k=0}^{2M} \binom{2M}{k} B_k^M(s) \int_N^\infty \frac{z^t}{(a+t)^{s+2M}} ((a+t)\log(z))^k dt \right|$$

with  $B_k^M(s) = (k+1-2M-s)_{2M-k}$ . We apply the usual upper bound for  $|\tilde{B}_n(t)| < 4n!/(2\pi)^n$  and formally interchange integration and the expansion of  $U(-2M, -2M+1-s, -(a+N)\log(z))$  given by

$$U(-2M, -2M+1-s, -(a+N)\log(z)) = \sum_{k=0}^{2M} \binom{2M}{k} B_k^M(s) ((a+t)\log(z))^k.$$

The integral above can be expressed in terms of the incomplete Gamma function  $\Gamma(a, z)$  as follows (similar to (9))

$$\int_N^\infty \frac{z^t}{(a+t)^{s+2M}} ((a+t)\log(z))^k dt = \frac{\log^k(z)}{z^a} (-\log(z))^{-k-1+2M+s} \\ \times \Gamma(k+1-2M-s, -(a+N)\log(z)) \quad (24)$$

Thus, we have

$$\sum_{k=0}^{2M} \binom{2M}{k} B_k^M(s) \int_N^\infty \frac{z^t}{(a+t)^{s+2M}} ((a+t)\log(z))^k dt \\ = \frac{\Gamma(1-s)}{z^a} \sum_{k=0}^{2M} \binom{2M}{k} Q(k+1-2M-s, -(a+N)\log(z)) \frac{(-\log(z))^{-k-1+2M+s}}{\log^{-k}(z)},$$

where we use  $B_k^M(s) = \Gamma(1-s)/\Gamma(k+1-2M-s)$ . Note that for  $s \in \mathbb{N}$  we take equation (24) to avoid the pole.  $\square$

The bound given in Theorem 2.2 give us a notably tight approximation of remainder (11). However, for large  $M$  the direct evaluation of the terminating series in (23) might be substantially expensive, being a not negligible part of the total computation time, therefore approximations for large order will be considered in Section 3.

## 2.2 Uniform asymptotic expansion for $\Phi(z, s, a)$

A suitable Laplace-type integral representation of  $\Phi(z, s, a)$  amenable to derive multiple asymptotic expansions [5], is given by

$$\Phi(z, s, a) = \frac{1}{\Gamma(s)} \int_0^\infty \frac{t^{s-1} e^{-at}}{1 - ze^{-t}} dt, \quad \Re(s) > 1, \Re(a) > 0, z \notin [1, \infty), \quad (25)$$

which serves to define the analytic continuation of the Lerch-series to  $z \in \mathbb{C} \setminus [1, \infty)$ . This integral has been chosen as starting point to derive asymptotic expansions for either large or small  $a$  (assuming that  $s$  and  $z$  are fixed) or for large  $z$  in [9], and for a Bernoulli-series representation as in [5]. The aim of this subsection is to extend the domain of computation of the Poincaré type asymptotic expansion for large  $a$  defined in [9] by constructing a uniform asymptotic expansion for large  $a$ ,  $s$  and  $z$ .

We proceed to construct that expansion by using the vanishing saddle point method described in [29]. This method is fundamentally a modification of Laplace's method applicable to integrals of the form

$$F_\lambda(z) = \frac{1}{\Gamma(\lambda)} \int_0^\infty t^{\lambda-1} e^{-zt} f(t) dt, \quad (26)$$

with  $\Re(\lambda) > 0$  and  $z$  large, in which  $\lambda$  might also be large. The resulting expansion is given by

$$F_\lambda(z) \sim \sum_{k=0}^{\infty} \frac{a_k(\mu) P_k(\lambda)}{z^{k+\lambda}},$$

where  $a_k(\mu)$  are the coefficients of the expansion of  $f(t)$  at the saddle point  $\mu = \lambda/z$  and coefficients  $P_k(\lambda)$  are expressible in terms of generalized Laguerre polynomials defined by

$$P_k(\lambda) = k! L_k^{-k-\lambda}(-\lambda). \quad (27)$$

At this point, we briefly recall the definition of the Eulerian polynomial and its connection with the polylogarithm function before stating the next proposition.

The Eulerian polynomial is defined as

$$A_k(z) = \sum_{j=0}^{k-1} \left\langle \begin{matrix} k \\ j \end{matrix} \right\rangle z^j, \quad (28)$$

where  $\left\langle \begin{matrix} k \\ j \end{matrix} \right\rangle$  are the Eulerian numbers [12]. The Eulerian polynomials satisfy the recurrence equation

$$A_0(z) = 1, \quad A_k(z) = \sum_{j=0}^{k-1} \binom{k}{j} A_j(z) (z-1)^{k-1-j}, \quad k \geq 1. \quad (29)$$

The Eulerian polynomial and polylogarithm are related by the functional equation

$$A_k(z) = \frac{(1-z)^{k+1}}{z} \text{Li}_{-k}(z) = \frac{(1-z)^{k+1}}{z} \sum_{j=1}^{\infty} j^k z^j, \quad |z| < 1, \quad (30)$$

and if  $|z| > 1$  then  $A_k(z) = (-1)^{k+1} A_k\left(\frac{1}{z}\right)$ .

**Proposition 2.3** For  $a, s, z \in \mathbb{C}$ ,  $\Re(a) > 0$  and  $z \notin [1, \infty)$  we have the following uniform asymptotic expansion for  $\Phi(z, s, a)$

$$\Phi(z, s, a) = \frac{e^\mu}{e^\mu - z} \left( \frac{1}{a^s} + \sum_{k=2}^{K-1} (-1)^k \frac{P_k(s)}{k! a^{k+s}} r^k A_k \left( \frac{e^\mu}{z} \right) \right) + \varepsilon_K(z, s, a), \quad (31)$$

where  $r = z/(e^\mu - z)$ .

**Proof:** We take  $f(t) = (1 - ze^{-t})^{-1}$  and  $\mu = s/a$  in (26), where  $\mu$  is the saddle point of the dominant part of the integral. Following closely the derivation in [9], we expand  $f(t)$  at  $t = \mu$  to obtain the Taylor expansion

$$f(t) = \sum_{k=0}^{\infty} a_k(\mu)(t - \mu)^k, \quad a_k(\mu) = (-1)^k \frac{e^\mu z^k}{k!(e^\mu - z)^{k+1}} \sum_{j=0}^{k-1} \langle k \rangle \left\langle j \right\rangle \left( \frac{e^\mu}{z} \right)^j. \quad (32)$$

After performing a few algebraic manipulations we obtain the final representation for the vanishing point expansion for  $\Phi(z, s, a)$ .  $\square$

We can clearly observe that for large values of  $\Re(s)$  and  $|z|$ , the asymptotic convergence of the expansion improves. Furthermore, from a numerical perspective, moderate to large values of  $\Re(s) < 0$  permit the evaluation of  $A_k(e^\mu/z)$  via the convergent series (30).

It remains to bound the error term in the expansion after truncation at  $k = K - 1$ . Let us consider the  $k$ -th term of expansion (31) defined as

$$|t_k| = \left| \frac{P_k(s)}{k! a^{k+s}} \left( \frac{z}{e^\mu - z} \right)^k A_k \left( \frac{e^\mu}{z} \right) \right| \leq \left| \frac{1}{k! a^{k+s}} \left( \frac{z}{e^\mu - z} \right)^k \right| |P_k(s)| \left| A_k \left( \frac{e^\mu}{z} \right) \right|.$$

A bound for the error term by comparison with a geometric series yields

$$\left| \sum_{k=K}^{\infty} t_k \right| \leq \frac{|t_K|}{1 - C}, \quad C = \left| \frac{t_{K+1}}{t_K} \right|, \quad (33)$$

iff  $C < 1$ , where  $t_K$  is the first omitted term in the expansion and

$$|\varepsilon_K(z, s, a)| \leq \left| \frac{e^\mu}{e^\mu - z} \right| \frac{|t_K|}{1 - C}.$$

In order to provide an effective upper bound for  $|t_k|$ , we compute two saddle point bounds for polynomials  $P_k(z)$  and  $A_k(z)$ .

**Proposition 2.4** For  $k > 1$  and  $z \in \mathbb{C} \setminus \{1\}$  the Eulerian polynomials satisfy the following bound

$$|A_k(z)| \leq k! \left| (z - 1)e^{\phi(t_0)} \right|, \quad (34)$$

where

$$t_0 = \frac{W(e^k k z) - k}{z - 1} \quad \text{and} \quad \phi(t_0) = -k \log t_0 - \log \left( z - e^{(z-1)t_0} \right),$$

and  $W(x)$  is the Lambert-W function which solves  $W(x)e^{W(x)} = x$ .

**Proof:** An integral representation for the Eulerian polynomials is obtained after applying Cauchy's integral formula to the exponential generating function given by

$$\sum_{k=0}^{\infty} A_k(z) \frac{t^k}{k!} = \frac{z - 1}{z - e^{(z-1)t}} \implies A_k(z) = \frac{k!(z - 1)}{2\pi i} \oint \frac{t^{-k-1}}{z - e^{(z-1)t}} dt,$$

which can be written in the form

$$A_k(z) = \frac{k!(z-1)}{2\pi i} \oint \frac{e^{\phi(t)}}{t} dt, \quad \phi(t) = -k \log t - \log(z - e^{(z-1)t}).$$

We compute the saddle point of the integrand by solving the following equation

$$\phi'(t) = \frac{(z-1)e^{(z-1)t}}{z - e^{(z-1)t}} - \frac{k}{t}, \quad t_0 = \frac{W(e^k k z) - k}{z-1}. \quad (35)$$

The principal contribution of the saddle point bound is obtained by substituting  $t_0$  into the integrand

$$|A_k(z)| \leq \frac{k!}{2\pi i} \left| (z-1)e^{\phi(t_0)} \right| \oint \frac{dt}{t} = k! \left| (z-1)e^{\phi(t_0)} \right|.$$

Finally, by the residue theorem we obtain the result.  $\square$

A similar analysis is carried out for polynomials  $P_k(z)$ . The use of the generating function for generalized Laguerre polynomials gives the Cauchy-type integral representation

$$P_k(s) = k! L_k^{-k-s}(-s) = \frac{k!}{2\pi i} \int_{\mathcal{C}} (1-t)^{k+s-1} e^{ts/(1-t)} \frac{dt}{t^{k+1}}, \quad (36)$$

where  $\mathcal{C}$  is a circle around the origin with a radius less than unity.

**Proposition 2.5** For  $k > 1$  and  $s \in \mathbb{C} \setminus \{0, 1\}$  the polynomials  $P_k(s)$  satisfy the following bound

$$|P_k(s)| \leq k! \left| e^{\phi(t_0)} \right|, \quad (37)$$

where

$$t_0 = \frac{\sqrt{k^2 + 4ks - 2k + 1} + k + 1}{2(1-s)}$$

and

$$\phi(t_0) = s \frac{t_0}{1-t_0} + (k+s-1) \log(1-t_0) - k \log t_0.$$

**Proof:** A proof follows the steps presented previously.  $\square$

Combination of both bounds (34) and (37) gives the final form for the error bound. The selection of the appropriate truncation point  $K$  to achieve a desired level of precision is detailed in Section 3.

### 2.3 Asymptotic expansion for large $z$

A careful reader shall have noticed that none of the previous series expansions are suitable for arbitrarily large  $\Re(z) > 0$ . The expansion in this subsection complements the asymptotic expansion described in [9] for  $z \in \mathbb{C} \setminus [0, \infty)$ ,  $\Re(a) > 0$  and  $\Re(s) > 0$  for large  $z$  and fixed  $a$  and  $s$

**Theorem 2.6** For  $(a, b, z) \in \mathbb{C}$  and  $\Re(a) > 0$  we have an asymptotic expansion for large  $a$  and  $z$ , and fixed  $s$  is given by

$$\begin{aligned} \Phi(z, s, a) &\sim \frac{1}{2a^s} + \frac{(-\log(z))^{s-1}}{z^a} \Gamma(1-s, -a \log(z)) \\ &+ \frac{1}{2a^s} \left( \frac{2}{\log(z)} - \coth\left(\frac{\log(z)}{2}\right) \right) \\ &+ \frac{1}{a^s} \sum_{k=1}^{\infty} \frac{(s)_k}{a^k (2\pi)^{k+1}} \left( \frac{1}{u^{k+1}} - \frac{\pi^{k+1}}{k!} \coth(\pi u)^{k-1} \operatorname{csch}(\pi u)^2 \mathcal{P}_k(\operatorname{sech}(\pi u)^2) \right), \end{aligned} \quad (38)$$

where  $u = \frac{\log(z)}{2\pi}$  and  $\mathcal{P}_k(x)$  are peak polynomials [28].

**Proof:** We start from the integral representation (13). Application of the binomial theorem yields

$$I = \Im \left( \int_0^\infty \frac{z^{-it}}{(1-it/a)^s} \frac{dt}{e^{2\pi t} - 1} \right) = \Im \left( \sum_{k=0}^\infty \frac{(s)_k}{k!} \left( \frac{i}{a} \right)^k \int_0^\infty \frac{z^{-it} t^k}{e^{2\pi t} - 1} dt \right).$$

Let us focus on the inner integral  $I_k$  defined as

$$I_k = \int_0^\infty \frac{z^{-it} t^k}{e^{2\pi t} - 1} dt = \int_0^\infty \frac{z^{-it} t^k (1 - e^{-t})}{(1 - e^{-t})(e^{2\pi t} - 1)} dt.$$

Noting that  $(1 - e^{-t})/(e^{2\pi t} - 1) = \frac{1}{2}(\coth(\pi t) - 1)(\sinh(t) - \cosh(t) + 1)$ , we split the integral obtaining a closed form in terms of the Hurwitz zeta function

$$I_k = k! \left( \frac{\zeta(k+1, i \log(z)/(2\pi))}{(2\pi)^{k+1}} - \frac{1}{(i \log(z))^{k+1}} \right) = \frac{k!}{(2\pi)^{k+1}} \zeta \left( k+1, 1 + i \frac{\log(z)}{2\pi} \right).$$

For  $k = 0$ ,  $\zeta \left( k+1, 1 + i \frac{\log(z)}{2\pi} \right)$  has a pole, so we proceed as follows

$$\Im \left( \int_0^\infty \frac{z^{-it}}{e^{2\pi t} - 1} dt \right) = - \int_0^\infty \frac{\sin(\log(z)t)}{e^{2\pi t} - 1} dt = \frac{1}{4} \left( \frac{2}{\log(z)} - \coth \left( \frac{\log(z)}{2} \right) \right).$$

Combining terms give us the asymptotic expansion for integral (13)

$$I \sim \frac{1}{4} \left( \frac{2}{\log(z)} - \coth \left( \frac{\log(z)}{2} \right) \right) + \sum_{k=1}^\infty \frac{(s)_k}{a^k (2\pi)^{k+1}} \Im \left( i^k \zeta \left( k+1, 1 + i \frac{\log(z)}{2\pi} \right) \right).$$

Hereinafter we use  $u = \frac{\log(z)}{2\pi}$  to simplify notation. Let us define the terms  $C_k(u)$  as

$$C_k(u) = \Im \left( i^k \zeta \left( k+1, 1 + iu \right) \right) = \frac{i^{k+1}}{2} \left( (-1)^k \zeta \left( k+1, 1 - iu \right) - \zeta \left( k+1, 1 + iu \right) \right),$$

where we remove the imaginary part. In order to eliminate the computations on the complex plane for real  $z$ , we expand<sup>1</sup>  $C_k(u)$  reducing compound arguments. The first five coefficients  $c_k(u) = 2C_k(u)$  are

$$\begin{aligned} c_1(u) &= \frac{1}{u^2} - \pi^2 \operatorname{csch}(\pi u)^2, \\ c_2(u) &= \frac{1}{u^3} - \pi^3 \coth(\pi u) \operatorname{csch}(\pi u)^2, \\ c_3(u) &= \frac{1}{u^4} - \frac{\pi^4}{6} \left( 4 \coth(\pi u)^2 \operatorname{csch}(\pi u)^2 + 2 \operatorname{csch}(\pi u)^4 \right), \\ c_4(u) &= \frac{1}{u^5} - \frac{\pi^5}{24} \left( 8 \coth(\pi u)^3 \operatorname{csch}(\pi u)^2 + 16 \coth(\pi u) \operatorname{csch}(\pi u)^4 \right), \\ c_5(u) &= \frac{1}{u^6} - \frac{\pi^6}{120} \left( 16 \coth(\pi u)^4 \operatorname{csch}(\pi u)^2 + 88 \coth(\pi u)^2 \operatorname{csch}(\pi u)^4 + 16 \operatorname{csch}(\pi u)^4 \right). \end{aligned}$$

From the observation of previous coefficients, we state the following identity, which proof follows by induction

$$\begin{aligned} C_k(u) &= \frac{1}{2u^{k+1}} - \frac{\pi^{k+1}}{2k!} \sum_{j=0}^{\lceil k/2 \rceil - 1} P(k, j) \coth(\pi u)^{k-1-2j} \operatorname{csch}(\pi u)^{2(j+1)} \\ &= \frac{1}{2u^{k+1}} - \frac{\pi^{k+1}}{2k!} \coth(\pi u)^{k-1} \operatorname{csch}(\pi u)^2 \sum_{j=0}^{\lceil k/2 \rceil - 1} P(k, j) \operatorname{sech}(\pi u)^{2j} \\ &= \frac{1}{2u^{k+1}} - \frac{\pi^{k+1}}{2k!} \coth(\pi u)^{k-1} \operatorname{csch}(\pi u)^2 \mathcal{P}_k(\operatorname{sech}(\pi u)^2). \end{aligned}$$

<sup>1</sup>We employ `FunctionExpand` in Mathematica [30].

where  $P(k, j)$  denotes the number of permutations of  $k$  numbers with  $j$  peaks, also known as *peak number* or *pk-number*, and  $\mathcal{P}_k(x)$  a *pk-polynomial*.  $\square$

Peak numbers  $P(k, j)$  give the sequence A008303 of the OEIS [27]. For  $k \geq 1$  and  $0 \leq j \leq k$ , we have a functional recursion generating a triangular array

$$P(k, j) = 2(j+1)P(k-1, j) + (k-2j)P(k-1, j-1). \quad (39)$$

Note that  $P(k, j) = 0$  for  $j \geq k/2$  and therefore  $\deg P_k(x) = \lceil k/2 \rceil - 1$ . Peak polynomials are given by the generating function for peak numbers  $P(k, j)$ .

$$\mathcal{P}_k(x) = \sum_{j=0}^{\lceil k/2 \rceil - 1} P(k, j)x^j.$$

Note that for values of  $|x| \rightarrow 1$  we can estimate its magnitude by the finite sum of peak numbers, since  $\sum_{j=0}^{\lceil k/2 \rceil - 1} P(k, j) = k!$ , hence

$$|\mathcal{P}_k(x)| \sim k!, \quad |x| \rightarrow 1. \quad (40)$$

The bivariate exponential generating function can be defined as in [10], [31]

$$\begin{aligned} \sum_{k=0}^{\infty} \mathcal{P}_k(p) \frac{x^k}{k!} &= 1 - \frac{1}{p} + \frac{\sqrt{p-1}}{p} \tan \left( x\sqrt{p-1} + \arctan(1/\sqrt{p-1}) \right) \\ &= \frac{\sqrt{1-p} \cosh(x\sqrt{1-p})}{\sqrt{1-p} \cosh(x\sqrt{1-p}) - \sinh(x\sqrt{1-p})} = \frac{1}{\sqrt{1-p} \coth(x\sqrt{1-p}) - 1} \end{aligned}$$

As customary in analytic combinatorics, application of Cauchy's integral formula to the bivariate exponential generating function gives

$$\mathcal{P}_k(p) = \frac{k!}{2\pi} \int_0^{2\pi} \frac{e^{-ikt}}{\sqrt{1-p} \coth(e^{it}\sqrt{1-p}) - 1} dt.$$

A remarkable result from the theory of enriched  $P$ -partitions is the functional relation between peak polynomials and Eulerian polynomials stated in [28]

$$\mathcal{P}_k \left( \frac{4x}{(1+x)^2} \right) = \frac{2^{k-1}}{(1+x)^{k-1}} A_k(x), \quad (41)$$

which allows us to use the upper bound in (34) to estimate the truncation point in (38). Furthermore, a good asymptotic estimate of  $P_k(x)$  for large order  $k$  can be derived from a Mittag-Leffler type decomposition of Eulerian polynomials [4]:

$$A_k(z) = C(k, z) \left( \frac{1}{\log(z)^{k+1}} + \sum_{j=1}^{\infty} \frac{1}{(\log(z) + 2\pi ij)^{k+1}} + \frac{1}{(\log(z) - 2\pi jk)^{k+1}} \right),$$

where

$$C(k, z) = \frac{e^{\pi i(k-1)}(1-z)^{k+1}k!}{z}.$$

Taking the prefactor of the expansion and applying the functional relation (41) we have

$$\mathcal{P}_k(x) \sim \frac{2^{k-1}k!e^{\pi i(k-1)}(1-u)^{k+1}}{(1+u)^{k-1}u \log(u)^{k+1}}, \quad u = \frac{2-x-2\sqrt{1-x}}{x}, \quad k \rightarrow \infty.$$

### 3 Algorithmic details and implementation

In this section we discuss in detail the implementation aspects and several proposed heuristics easy to evaluate while being effective in practice. All algorithms described are implemented in Python<sup>2</sup> using the mpmath library for arbitrary-precision floating-point arithmetic [20] with GMPY2, which supports integer and rational arithmetic via the GMP library [13] and real and complex arithmetic by the MPFR [11] and MPC [7] libraries.

As it is well-known, numerical evaluation of special functions requires the use of several methods of computation to cover the whole regime of the parameters. We aim to sketch the building blocks of a basic algorithm, which have been tested to work reasonable well for most cases, but we do not dare to claim that it will cover the whole function's domain optimally. For those cases either not covered by current series expansions or prone to numerical instability, we select numerical complex integration, which serves as a backup method.

#### 3.1 Evaluation of L-series

The L-series of the form (1) are in general difficult to accelerate due to the non recursive scheme of computation. In order to employ common acceleration techniques such as parallelization, the determination of the optimal truncation level is crucial. As described in the previous section, a bound for the remainder term of the L-series can be constructed as follows

$$\left| \sum_{k=K}^{\infty} \frac{z^k}{(k+a)^s} \right| \leq \frac{|z|^K}{|(K+a)^s|(1-C_K(z,s,a))} \leq \frac{|z|^K}{|(K+a)^s|(1-|z|)}, \quad (42)$$

where

$$C_K(z,s,a) = \left| \frac{z(K+a)^s}{(K+1+a)^s} \right|, \quad \lim_{K \rightarrow \infty} C_K(z,s,a) = |z|. \quad (43)$$

The required number of terms  $K$  to obtain a result with  $P$ -bit precision can be obtained by performing a simple linear search, which is generally sufficient to target an absolute error of about  $2^{-P}$ . However, a more efficient approximation of  $K$  is yielded by solving the following equation with the first omitted term,  $z^K(K+a)^{-s} = 2^{-P}$  for  $K$ . The first solution in closed form is given by

$$K = -\frac{sW(\phi(z,s,a,P)) + a \log(z)}{\log(z)}, \quad \phi(z,s,a,P) = -\frac{(2^{-P}z^a)^{-1/s} \log(z)}{s}.$$

We distinguish two different approximations for  $K$ , denoted as  $\hat{K}$ , depending on  $\Re(s)$ . For  $\Re(s) > 0$

$$\hat{K} = \left\lceil \left\lfloor \frac{\Re(s)W_0(\phi_1(z,s,a,P)) + |a| \log(|z|)}{\log(|z|)} \right\rfloor \right\rceil,$$

where  $[x]$  denotes the nearest integer function and

$$\phi_1(z,s,a,P) = -\frac{(2^{-P}|a|^{-\Re(s)-1}|z|^{|a|})^{-1/(\Re(s)+1)} \log(|z|)}{\Re(s)+1}.$$

For  $\Re(s) < 0$

$$\hat{K} = \left\lceil \left\lfloor \frac{\Re(s)W_{-1}(\phi_2(z,s,a,P)) + |a| \log(|z|)}{\log(|z|)} \right\rfloor \right\rceil,$$

where

$$\phi_2(z,s,a,P) = -\frac{(2^{-P}|a|^{-\Re(s)}|z|^{|a|})^{-1/\Re(s)} \log(|z|)}{\Re(s)}.$$

<sup>2</sup><https://sites.google.com/site/guillermonavaspalencia/software/lerch.py>

Given that  $\phi_1, \phi_2 \in \mathbb{R}$ , we use the principal branch  $W_0(z) = W(z)$  when  $\Re(s) > 0$ , since  $\phi_1 > 0$  and the branch  $W_{-1}(z)$  for  $\Re(s) < 0$ , since  $\phi_2 \in (-1/e, 0)$ . Remember that  $W(x)$  is two-valued for  $-1/e \leq x < 0$ . Numerical tests suggest that these approximations for choosing  $K$  are sufficient to obtain good estimates of the required number of terms. We note that  $\hat{K}$  can be computed using 53-bit machine floating-point arithmetic.

Several heuristics are implemented to compensate catastrophic cancellation for cases when  $\Re(z) < 0$  and/or  $\Re(s) < 0$ . In particular, for  $\Re(s) < 0$  we increase the working precision  $P^W = P + \lfloor P/3 \rfloor + \lceil -\Re(s) \rceil$ . On the other hand, for the case  $\Re(s) > 1$  and  $z \in \mathbb{R}_{<0}$  we employ the linear acceleration methods for alternating series described in [3]. This method is used when  $\hat{K} > 1.2\lceil 1.31D \rceil$ , where  $D$  is the precision digits. For all other cases, we add up to 20 guard bits to the working precision.

The computation of the L-series is particularly simple to parallelize by assigning a block of size  $k : k \leq N$  to each thread. This parallelization scheme is implemented using the `multiprocessing` module in Python. Based on experiments, parallelization provides a significant speedup factor for  $\hat{K} > 1024$  or  $P \geq 1024$  bits.

We remark that L-series converges rather slowly when  $|z| \rightarrow 1$ . It is possible to employ convergence acceleration techniques to obtain an efficient evaluation of the Lerch transcendent; see the application of combined nonlinear-condensation transformation in [16]. Alternatively, the Euler-Maclaurin formula is also convenient for those cases, as shown later.

## 3.2 Evaluation of Euler-Maclaurin formula

### 3.2.1 Evaluation of the error bound

For a precision of  $D$  digits, we choose  $N = \lfloor D/3 \rfloor$ . For large  $\Re(a) > 0$  we choose  $N = 0$  if the following condition is satisfied

$$\Re(a) > |\Re(s)| + |\Re(z)| + D.$$

The number of terms  $M$  can be effectively approximated by solving  $t_k = 2^{-P}$ , where  $P$  is the precision in bits and  $t_k$  is the asymptotic estimate in (22), which yields

$$M \sim \left\lceil \frac{1}{2} \left| \frac{\log(2^{-P-1} \log(z))}{\log(2\pi) - \log(\log(z))} \right| \right\rceil.$$

This is a near-optimal approximation at high-precision. In practice, the asymptotic estimate of  $M$  is used for  $P \geq 500$ , otherwise we use the heuristic  $M = N + \lfloor P/3 \rfloor$ . There is an unavoidable trade-off when choosing  $N$  and  $M$ , large values results in catastrophic cancellation since the L-series might be unstable, especially for  $z \in \mathbb{C} \setminus \mathbb{R}$ , but reduces the number of terms  $M$ , therefore the time spent computing Bernoulli numbers, which represents a significant amount of the total computation time.

We can evaluate the coefficient in the error bound (23) using a recurrence. Computation of  $Q(k+1-2M-s, -(a+N)\log(z))$  only requires the initial value  $Q(1-s, -(a+N)\log(z))$ , which can be computed re-using  $\Gamma(1-s, -(a+N)\log(z))$  in (10). Subsequent terms can be computed at lower precision via the recurrence

$$Q(a, z) = Q(a-1, z) + \frac{e^{-z}z^{a-1}}{\Gamma(a)}, \quad a \in \mathbb{C} \setminus \mathbb{Z}^-,$$

or

$$\Gamma(a, z) = (a-1)\Gamma(a-1, z) + e^{-z}z^{a-1}, \quad a \in \mathbb{C}.$$

A recurrence for the rest of terms in the coefficients is trivial. The direct evaluation of the recurrence requires  $\mathcal{O}(M)$  arithmetic operations, therefore the associated computational cost is not negligible for very large  $M$ , as previously mentioned.

For large  $M$  we might use asymptotic estimates to reduce the complexity, for example the case  $M > |(a + N) \log(z)|$

$$\begin{aligned} |R| &\sim \frac{4}{(2\pi)^{2M}} \left| (-2M + 1 - s) {}_{2M}E_{2M+s}(- (a + N) \log(z)) \frac{(a + N)^{-2M-s+1}}{z^{a+N}} \right| \\ &\sim \frac{4}{(2\pi)^{2M}} \left| \frac{(-2M + 1 - s) {}_{2M}}{2M + s} \frac{(\log(z)/z)^{a+N}}{(a + N)^{2M+s-1}} \right|, \end{aligned} \quad (44)$$

where  $E_\nu(z)$  is the generalized exponential integral [24] and we take the asymptotic estimate  $E_\nu(z) \sim e^{-z}/\nu$  as  $\nu \rightarrow \infty$ .

For  $M \sim |(a + N) \log(z)|$  and  $|s| \ll M$ , we use the first order estimate of the Franklin-Friedman expansion for  $U(a, b, z)$  in [25] given by

$$U(-2M, -2M + 1 - s, -(a + N) \log(z)) \sim \left( 1 + \frac{2M}{(a + N) \log(z)} \right)^{-s} ((a + N) \log(z))^{2M},$$

replacing it in (23) and after observation that the remaining integral is expressible in terms of the incomplete gamma function we obtain

$$|R| \sim \frac{4}{(2\pi)^{2M}} \left| \frac{(-\log(z))^{2M+s-1}}{3^{\min(\Re(s), 0)} z^a} \Gamma(1 - s, -(a + N) \log(z)) \right|. \quad (45)$$

Table 1 shows a few examples when  $|z| < 1$ , otherwise the integral (11) is not well defined. Approximations (44) and (45), although being quite simple, might be used to compute a crude estimate of the magnitude of the remainder in reasonable time.

$z$	$s$	$a$	$N$	$M$	(11)	(23)	(44)	(45)
0.8	3.2	10.5	6	15	$7.5e-30$	$1.5e-28$	$1.3e-38$	$1.1e-47$
$0.5 + 0.2i$	$-30.2 - i$	$10.5 + 5i$	10	40	$4.8e-24$	$1.0e-22$	$8.1e-37$	$3.2e-20$
$0.5 + 0.2i$	$-30.2 - i$	$100.5 + 5i$	100	2000	$2.5e+302$	$5.6e+302$	$3.9e+282$	–
$0.5 + 0.7i$	$-3.2 + 10i$	$10.5 + 5i$	250	300	$1.6e-503$	$4.8e-502$	$1.5e-503$	$5.4e-496$
$0.5 + 0.7i$	$30.2 + 10i$	$10.5 + 10.5i$	600	700	$9.9e-1240$	$1.2e-1238$	$7.5e-1216$	$3.8e-1264$

Table 1: Effectiveness of bound (23) in error term of the Euler-Maclaurin formula.

### 3.2.2 Evaluation of the tail

A more interesting form of the tail (10) is obtained by applying Kummer's transformation to  $U(-2k + 1, -2k + 2 - s, -(a + N) \log(z))$ , thus

$$T = z^N \left( \frac{1}{2(a + N)^s} + (-\log(z))^s \sum_{k=1}^M \frac{B_{2k}}{(2k)!} (-\log(z))^{2k-1} U(s, s + 2k, -(a + N) \log(z)) \right).$$

For this particular case,  $U(a, b, z)$  reduces to a polynomial in  $-(a + N) \log(z)$  of degree  $2k - 1$ , indeed expressible in terms of generalized Laguerre polynomials, given by

$$U(s, s + 2k, -(a + N) \log(z)) = (- (a + N) \log(z))^{-s} \underbrace{\sum_{j=0}^{2k-1} \binom{2k-1}{j} \frac{(s)_j}{(- (a + N) \log(z))^j}}_{T_2^{(k)}}.$$

Terms  $T_2^{(k)}$  can be constructed using a linear holonomic recurrence equation. Let us define the constants expressions  $p$  and  $q$

$$p = s - (a + N) \log(z), \quad q = -\frac{1}{(a + N) \log(z)}.$$

The sequence of terms  $T_2^{(k)}$  satisfy the recurrence equation

$$\begin{aligned} T_1^{(k)} &= [pT_2^{(k-1)} + (2k-3)(T_2^{(k-1)} - T_1^{(k-1)})]q \\ T_2^{(k)} &= [pT_1^{(k)} + (2k-2)(T_1^{(k)} - T_2^{(k-1)})]q \end{aligned}$$

for  $k \geq 2$ , with initial values

$$T_1^{(1)} = 1, \quad T_2^{(1)} = 1 - \frac{s}{(a+N)\log(z)}.$$

A matrix form for  $k \geq 2$  is defined as

$$\begin{pmatrix} T_1^{(k)} \\ T_2^{(k)} \\ T_2^{(k-1)} \end{pmatrix} = \begin{pmatrix} q(p+2k-3) & q(3-2k) & 0 \\ q(2-2k) & 0 & q(p+2k-2) \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} T_2^{(k-1)} \\ T_1^{(k-1)} \\ T_1^{(k)} \end{pmatrix}$$

or simply

$$\begin{pmatrix} T_2^{(k)} \\ T_1^{(k)} \end{pmatrix} = q \begin{pmatrix} k(4k+4p-12) + (p-5)p+8 & k(-4k-2p+10) + 3p-6 \\ 3-2k & p+2k-3 \end{pmatrix} \begin{pmatrix} T_2^{(k-1)} \\ T_1^{(k-1)} \end{pmatrix}.$$

The complexity of the recurrence scheme is  $\mathcal{O}(MP)$  and requires a small temporary storage. Note that a matrix recurrence for the sequence of coefficients  $T_*^{(k)}$  is suitable in a binary splitting scheme. The previous analysis results in a more tractable expression for the tail  $T$

$$T = \frac{z^N}{(a+N)^s} \left( \frac{1}{2} + \sum_{k=1}^M \frac{B_{2k}}{(2k)!} (-\log(z))^{2k-1} T_2^{(k)} \right). \quad (46)$$

Now the terms of tail sum  $T$  satisfy a recurrence equation except for the multiplication by Bernoulli numbers. The Bernoulli numbers are cached for repeated evaluation, but computing them the first time at very high precision is time-consuming. We do not attempt to improve current implementations but rather rely on the algorithm implemented in `mpmath`, which automatically caches Bernoulli numbers  $B_n$  when  $n < 3000$  for multiple evaluations. For larger values of  $n$  the connection to Riemann zeta function  $\zeta(n)$  is used. Many recursive algorithms for computing  $B_0, \dots, B_n$  such as  $B_n = -\sum_{k=0}^{n-1} \frac{B_k}{k!(n-k+1)}$  require  $\mathcal{O}(n^2)$  arithmetic operations. As an alternative, an algorithm based on recycling terms in the Riemann zeta function series expansion, which also have cubic complexity, is implemented in [18].

### 3.3 Evaluation of asymptotic expansions

The main drawback of the asymptotic expansions in (31) and (38) is the difficulty of computing a large number of Eulerian and peak polynomials efficiently. Computing the first  $k$  Eulerian polynomials simultaneously can be performed by using the recursion in (29). Thus, given  $A_0(z), \dots, A_{k-1}(z)$ , we can compute  $A_k(z)$  in  $\mathcal{O}(k)$  arithmetic operations, and noting that  $A_k(z)$  has  $\mathcal{O}(k \log k)$  bits from (34), the algorithm needs  $\mathcal{O}(k^{3+o(1)})$  bit operations and requires  $\mathcal{O}(k^2 \log k)$  space to store previous  $A_j(z), j < k$ . For example, using a straightforward implementation, we compute  $A_0(2), \dots, A_{1000}(2)$  at 333-bit precision in 1.51 seconds on a 2.6 GHz Intel i7 processor.

To compute  $k$  Eulerian polynomials in time complexity  $\mathcal{O}(k^{2+o(1)})$  we might apply a *multisectioning* scheme to the bivariate exponential generating function. Alternatively, it is possible to recycle terms of the sum (30) to speedup multievaluation, considering that terms  $j^k z^j$  can be optimized to only compute binary exponentiation when  $j$  is prime and multiplication otherwise. The required number of terms is approximated by solving  $J^k z^J = 2^{-P}$ ,

$$J^* \approx \left\lceil \left\lceil \Re \left( -\frac{kW_{-1} \left( -\frac{(2^{-P})^{1/k} \log(|z|)}{k} \right)}{\log(|z|)} \right) \right\rceil \right\rceil. \quad (47)$$

For large  $k$  the size of  $J^*$  growth rapidly, therefore it is convenient to apply asymptotic faster methods such as the Mittag-Leffler type decomposition introduced in §2.3, which acts as an asymptotic expansion. For  $z \in \mathbb{R}$ , two optimizations can be implemented:

$$A_k(z) = \frac{(z-1)^{k+1}k!}{z} \Re \left( \frac{1}{\log(z)^{k+1}} + 2 \sum_{j=1}^{\infty} \frac{1}{(\log(z) + 2\pi ij)^{k+1}} \right), \quad z \in \mathbb{R}^+,$$

$$A_k(z) = \frac{(z-1)^{k+1}k!}{z} \Re \left( \frac{1}{\log(z)^{k+1}} + \sum_{j=1}^{\infty} \frac{1}{(\log(z) - 2\pi ij)^{k+1}} \right), \quad z \in \mathbb{R}^-.$$

To compute a single  $A_{1000}(1/5)$  at 333-bit, the power series requires  $J^* = 5545$  whereas the Mittag-Leffler decomposition only needs  $J^* = 59$  terms, hence a complete algorithm shall combine the iterative computation via the three-term recurrence and the asymptotic expansion as  $k \rightarrow \infty$ .

Like other orthogonal polynomials, polynomials  $P_k(\lambda)$  in (27), which are strongly related to Tricomi-Carlitz polynomials, satisfy three-term recurrence,

$$P_0(\lambda) = 1, \quad P_1(\lambda) = 0, \quad \text{and} \quad P_{k+1}(\lambda) = k(P_k(\lambda) + \lambda P_{k-1}(\lambda)), \quad k > 1.$$

Given the complexity of computing a large number of Eulerian polynomials, the cost of the three-term recurrence is almost negligible.

The truncation level  $K$  in (33) is computed at lower precision via linear search and  $C$  is estimated as

$$C = \left| \frac{t_{K+1}}{t_K} \right| \sim \left| \frac{z}{z - e^\mu} \frac{(s+K)(z-1)}{a \log(z)} \right|, \quad k \rightarrow \infty,$$

from which we obtain an estimate of the number of terms

$$K_{max} \approx \left\lfloor \frac{a \log(z)(z - e^\mu)}{z(z-1)} - s \right\rfloor.$$

Computation of peak polynomials is carried out using the generating function for peak numbers for moderate  $k$ , which evaluation only involves roughly half of the terms  $k/2$  compared to the Eulerian polynomials. For example, computing the triangular array for the first 1000 peak numbers using recurrence (39) takes 1.55 seconds. For larger  $k$  the functional relation with the Eulerian polynomial is applied. A trickier aspect of the asymptotic expansion (38) is to determine the optimal truncation  $K$ . The coefficients  $c_k(u)$  behave as

$$|c_k(u)| = \left| \frac{1}{u^{k+1}} - \frac{\pi^{k+1}}{k!} \coth(\pi u)^{k-1} \operatorname{csch}(\pi u)^2 \mathcal{P}_k(\operatorname{sech}(\pi u)^2) \right| \sim \frac{1}{|1 - iu|^k},$$

as  $k \rightarrow \infty$ . Given the ratio of convergence of the asymptotic expansion, we can estimate the maximum number of terms  $K_{max}$ , thus the maximum attainable accuracy as follows

$$\left| \frac{t_{K+1}}{t_K} \right| \sim \left| \frac{(s+K)}{a2\pi(1-iu)} \right|, \quad k \rightarrow \infty, \quad K_{max} \approx |a(2\pi - i \log(z)) - s|.$$

It remains to estimate the required numbers of terms  $K$  to target  $P$ -bit accuracy, which is approximated heuristically and subsequently refined via linear search using

$$K \approx \left\lceil \left\lfloor \frac{\varphi(s)}{W_{-1}\left(\frac{\varphi(s)}{a2\pi \log(z)}\right)} \right\rfloor \right\rceil,$$

where  $\varphi(s) = -\log(2)^P - \log(1+s)$ . In fact, we slightly increase  $K$  by a factor  $\approx 1.2$ , which works well in practice. Hence, we can evaluate the asymptotic expansion as long as  $K \leq K_{max}$  to target an absolute error of  $2^{-P}$ .

### 3.4 Numerical integration

The current implementation in `mpmath` computes the Lerch transcendent via numerical integration using the double-exponential method for the integral representation (2) employing the `quad` function. We choose numerical integration for  $\Re(a) > 0$  as a backup method when computation by aforementioned methods is not satisfactory. Note that a few optimizations are possible for real parameters by rewriting the integrand, for example, the integral representation (13) when  $z > 0$  and  $a > 0$  or

$$I = -\Im \left( \int_0^\infty \frac{(1 - it/a)^s z^{it}}{(a^2 + t^2)^{s/2} (1 + t^2/a^2)^{s/2} (e^{2\pi t} - 1)} dt \right), \quad z, s, a \in \mathbb{R}, z > 0,$$

both integral representations avoiding evaluation of trigonometric functions. The computation at high-precision, say 1000 digits onwards, is generally costly compared to asymptotic methods, therefore this is the method of choice when only strictly indispensable.

## 4 Benchmark

In this Section, we benchmark our implementation to current state-of-the-art software supporting evaluation of the Lerch transcendent function to arbitrary-precision. Tests were conducted on an Intel(R) Core(TM) i7-6700HQ CPU at 2.60GHz, using up to 4 cores for parallel mode, running Ubuntu Linux. We compare the computing times of Mathematica 10.4 and `mpmath` 1.0.0 using functions `Timing[]` and `time.perf_counter()`, respectively. For `mpmath` we set the precision in bits  $p$  using `mpmath.mp.prec = p`, whereas for Mathematica the desired level of precision in digits  $d$  is set with `N[... , d]`, applying the conversion factor  $d = \lceil 0.301p \rceil$ . To assess the correctness of our implementation, we compare to Mathematica at higher precision since it is frequently faster and more reliable than `mpmath`. Note, however, that Mathematica attempts to achieve  $d$  digits of precision might fail unexpectedly, therefore we check the consistency of results at increasing levels of precision.

The following tables show timing results to compute the Lerch transcendent for various regimes of the parameters and argument, varying the level of precision. We remark that Mathematica and `mpmath` use GMP internally, so timing measurements are directly comparable.

Table 2 shows the performance of the Euler-Maclaurin formula (21) for small  $z$  and moderate values of  $s$  and  $a$ . The Euler-Maclaurin formula is implemented in a loop manner checking the level of cancellation at each iteration and increasing the working precision accordingly to correct it. Hence, a better estimation of the total amount of cancellation would reduce the computation time considerably. However, as we see for these cases, both Mathematica and `mpmath` are regularly an order of magnitude slower. Furthermore, as previously noted, larger values of  $|a|$  improve the convergence of the series, reducing significantly the number of terms  $N$  and  $M$ ; Table 3 shows the metrics corresponding to the last iteration.

$\Phi(z, s, a)$	bits	mpmath	Mathematica	Euler-Maclaurin	Parallel
	64	0.096 (0.139)	0.313	0.008 (0.013)	-
$z = 2.5 + 1.5i$	333	1.09 (1.21)	0.672	0.041 (0.089)	-
$s = 1.25 + 2i$	1024	8.39 (9.38)	3.14	0.128 (0.233)	-
$a = 3.5 + 5i$	3333	154.4 (161.1)	27	1.64 (2.37)	-
	10000	1564.6	438	25.04 (33.33)	19.06 (27.32)
	64	0.263 (0.295)	0.047	0.016 (0.039)	-
$z = 2.5 + 7.5i$	333	1.79 (1.98)	0.250	0.110 (0.250)	-
$s = -50.25 + 10i$	1024	6.59 (7.05)	1.58	0.72 (1.49)	-
$a = 1.5 - i$	3333	133.6 (135.1)	21.66	11.83 (16.64)	8.72 (13.22)
	10000	1453	409.1	190.3 (224.5)	153.2 (196.5)
	64	0.253 (0.324)	0.031	0.013 (0.022)	-
$z = 2.5 + 0.5i$	333	1.79 (1.92)	0.265	0.058 (0.104)	-
$s = -100.25 + 10i$	1024	12.66 (13.94)	7.72	0.298 (0.685)	-
$a = 100.5 - 10i$	3333	120.7 (127.1)	83.14	2.97 (4.19)	-
	10000	1500.6	> 1800	24.65 (32.89)	18.88 (28.34)

Table 2: Time (in seconds) to compute  $\Phi(z, s, a)$  with moderate values of  $z$ ,  $s$  and  $a$  to 64, 333, 1024, 3333 and 10000 bits of precision. First evaluation pre-computing Bernoulli numbers within parentheses. Maximum time 1800 seconds.

Current implementation does not incorporate complexity-reducing methods for the evaluation of the tail but simply uses the recurrence scheme and only includes optional parallelization of the truncated L-series, thus limiting the observable improvement by activating the parallel mode.

$\Phi(z, s, a)$	bits	N	M	$P^W$ (bits)
	64	7	32 (H)	86
$z = 2.5 + 1.5i$	333	39	172 (H)	400
$s = 1.25 + 2i$	1024	123	247 (A)	1229
$a = 3.5 + 5i$	3333	401	805 (A)	4000
	10000	1203	2416 (A)	12000
	64	28	122 (H)	84
$z = 2.5 + 7.5i$	333	127	402 (A)	383
$s = -50.25 + 10i$	1024	364	1146 (A)	1094
$a = 1.5 - i$	3333	1115	3503 (A)	3346
	10000	3193	10032 (A)	10419
	64	10	46 (H)	108
$z = 2.5 + 0.5i$	333	52	97 (A)	528
$s = -100.25 + 10i$	1024	156	285 (A)	1561
$a = 100.5 - 10i$	3333	480	876 (A)	4790
	10000	1214	2216 (A)	12108

Table 3: Number of terms  $N$ ,  $M$  in the Euler-Maclaurin expansion and working precision  $P^W$  for Euler-Maclaurin cases. (A) and (H) indicate the method used to estimate  $M$ , asymptotic and heuristic, respectively.

Table 4 assesses the performance of the L-series implementation and its particular cases for  $|z| < 1$  and small values of  $s$  and  $a$ . Due to the performance gap between Mathematica and mpmath (mpmath only implements numerical integration), only the former is used for benchmarking on subsequent tests. Results show that our implementation is comparable to Mathematica (presumably evaluating the same L-series) at lower precision and it is found to be surprisingly faster at higher precision<sup>3</sup>. Moreover, we observe that our parallelization scheme achieves speedup ratios close to theoretical maximum, which apparently is not imple-

<sup>3</sup>We guess the poor performance is due to incorrect error tracking, which overestimates the required working precision.

mented in Mathematica. Interestingly, the Euler-Maclaurin formula should be the preferred algorithm at low-medium precision when  $|z| \sim 1$ .

$\Phi(z, s, a)$	bits	Mathematica	L-series	Parallel	Euler-Maclaurin
	64	0.0011	0.0011	-	-
$z = 1/4$	333	0.0067	0.0055	-	-
$s = 10/4$	1024	0.0339	0.0155	-	-
$a = 20/7$	3333	0.7313	0.0660	0.0431	-
	10000	21.406	0.4885	0.1513	-
	64	0.0015	0.0069	-	0.0031 (0.0053)
$z = 9/10$	333	0.0109	0.0607	0.0429	0.0153 (0.0255)
$s = 10/4$	1024	0.0984	0.2165	0.1052	0.0166 (0.0222)
$a = 20/7$	3333	1.8843	1.0422	0.3281	0.0953 (0.1197)
	10000	39.937	8.6969	2.4881	2.7331 (3.1578)
	64	0.0031	0.0020	-	-
$z = -6/10$	333	0.0156	0.0083	-	-
$s = 10/4$	1024	0.1422	0.0243	-	-
$a = 20/7$	3333	3.0922	0.0983	-	-
	10000	21.2969	0.7500	-	-

Table 4: Time (in seconds) to compute  $\Phi(z, s, a)$  for small argument  $|z|$ .

The third example assesses the performance of the series acceleration technique for alternating series, which while it is hardly parallelizable, it is consistently faster than Mathematica for all tested instances.

Table 5 shows the time to compute the Lerch transcendent using the asymptotic expansion (38) for large  $z$  and  $a$ . The optimal truncation of the first test is  $K_{max} = 1598$ , limiting the evaluation at 3333 bits of precision, which would require  $K = 2767$ . The optimal truncation for the second test is  $K_{max} = 22297$  requiring up to 1952 terms at 10000 bit of precision. As noted, the time spent on the computation of a large number of peak polynomials accounts for a significant amount of the total time, therefore a more sophisticated and efficient algorithm would be needed at higher precision. On the other hand, for multiple evaluations, peak numbers can be cached same as Bernoulli numbers.

Numerical experiments show a performance deterioration of the Euler-Maclaurin formula as  $z$  increases due to catastrophic cancellation, therefore its use should be restricted to low precision calculations. Our implementation of the asymptotic expansion exhibits fast convergence for large parameters, but the limitation on the achievable accuracy forces a switch to numerical integration depending on the desired level of precision.

$\Phi(z, s, a)$	bits	mpmath	Mathematica	Euler-Maclaurin	Asymptotic	$K$	peak time
$z = 140$	64	0.0684	0.0154	0.0027	0.0022	9	6.3%
$s = 1/4$	333	0.7859	0.1219	2.2342	0.0134	59	10.9%
$a = 200$	1024	5.0361	0.7297	-	0.1931	254	10.5%
	64	0.1238	0.0661	-	0.0017	6	4.6%
$z = 10000$	333	1.5456	0.2078	-	0.0066	34	11.5%
$s = 10/4$	1024	9.9478	1.0406	-	0.0481	122	10.0%
$a = 2000$	3333	94.362	15.141	-	0.9498	493	8.4%
	10000	1978.2	283.21	-	39.779	1952	6.0%

Table 5: Time (in seconds) to compute  $\Phi(z, s, a)$  for large parameter  $a$  and argument  $z$ . Comparison to Euler-Maclaurin at low precision. The rightmost column shows the percentage of the total time devoted to computation of  $K$  peak numbers.

Finally, Table 6 compares the uniform asymptotic expansion (31) to the asymptotic expansion (38). Results show that the former expansion should be the preferred choice at low-medium precision for sufficiently large parameters and argument, otherwise the previous methods generally show superior performance. Note that the computation of Eulerian

polynomials accounts for the majority of the total time, consequently any improvement on this respect will directly reduce the reported timings.

$\Phi(z, s, a)$	bits	mpmath	Mathematica	Asymptotic	Uniform	Eulerian time
$z = -200.65$	64	0.0228*	0.0469	0.0031 (16)	0.0173 (20)	97.0%
$s = 100.25$	333	0.0149*	0.1563	0.0412 (104)	0.0592 (60)	97.5%
$a = 501.5$	1024	1.0219**	0.8438	0.7512 (421)	0.5151 (229)	98.8%
$z = -20000$	64	0.0101*	0.0312	0.0027 (15)	0.0051 (9)	93.2%
$s = 100.25$	333	0.0168*	0.1875	0.0362 (93)	0.0436 (53)	96.9%
$a = 501.5$	1024	1.7736	1.0313	0.5424 (365)	0.2964 (196)	98.3%

Table 6: Time (in seconds) to compute  $\Phi(z, s, a)$  for large parameter  $a$  and  $s$ , and argument  $z$ . Number of terms for each expansion within parentheses. For mpmath: (\*) and (\*\*) indicate no answer and inaccurate answer, respectively.

## 5 Discussion

The algorithms presented in this work are an important step towards a complete arbitrary-precision implementation of the Lerch transcendent using asymptotically fast methods. A fundamental improvement to our implementation is to devise a more intelligent strategy to address cancellation issues for the Euler-Maclaurin formula, which should yield a significant reduction of the current overhead factor.

Further work is needed to develop an efficient multithreaded implementation of the asymptotic expansions. More importantly, it remains an open problem whether there is a fast memory-efficient algorithm for computing a large number of Eulerian and peak polynomials.

## References

- [1] T. M. Apostol. On the lerch zeta function. *Pacific J. Math.*, 1(2):161–167, 1951.
- [2] H. Cohen. *Number Theory - Vol II: Analytic and Modern Tools*. Springer-Verlag - Graduate Texts in Mathematics 240, 2007.
- [3] H. Cohen, F. Rodriguez Villegas, and D. Zagier. Convergence acceleration of alternating series. *Experiment. Math.*, 9(1):3–12, 2000.
- [4] O. Costin and S. Garoufalidis. Resurgence of the fractional polylogarithms. 2009.
- [5] R. E. Crandall. Unified algorithms for polylogarithms,  $L$ -series, and zeta variants. 2012.
- [6] *NIST Digital Library of Mathematical Functions*. <http://dlmf.nist.gov/>, Release 1.0.14 of 2016-12-21. F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller and B. V. Saunders, eds.
- [7] Andreas Enge, Mickaël Gastineau, Philippe Théveny, and Paul Zimmermann. *mpc — A library for multiprecision complex arithmetic with exact rounding*. INRIA, 1.0.3 edition, February 2015. <http://mpc.multiprecision.org/>.
- [8] A. Erdélyi, W. Magnus, F. Oberhettinger, and F. G. Tricomi. *Higher Transcendental Functions. Vol. I*. McGraw-Hill Book Company, Inc., New York-Toronto-London, 1953.
- [9] C. Ferreira and J. L. López. Asymptotic expansions of the Hurwitz-Lerch zeta function. *Journal of Mathematical Analysis and Applications*, 298(1):210–224, 2004.
- [10] C. J. Fewster and D. Siemssen. Enumerating permutations by their run structure. *The electronic journal of combinatorics*, 21, 10 2014.

- [11] Laurent Fousse, Guillaume Hanrot, Vincent Lefèvre, Patrick Pélissier, and Paul Zimmermann. Mpf: A multiple-precision binary floating-point library with correct rounding. *ACM Trans. Math. Softw.*, 33(2), June 2007.
- [12] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 1994.
- [13] Torbjörn Granlund and the GMP development team. *GNU MP: The GNU Multiple Precision Arithmetic Library*. <http://gmplib.org/>.
- [14] A. Griffin, Wen-Chin Wu, and S. Stringari. Hydrodynamic modes in a trapped Bose gas above the Bose-Einstein transition. *Phys. Rev. Lett.*, 78:1838–1841, Mar 1997.
- [15] J. Guillera and J. Sondow. Double integrals and infinite products for some classical constants via analytic continuations of lerch’s transcendent. *The Ramanujan Journal*, 16(3):247–270, Aug 2008.
- [16] U. D. Jentschura, P. J. Mohr, G. Soff, and E. J. Weniger. Convergence acceleration via combined nonlinear-condensation transformations. *Computer Physics Communications*, 116(1):28 – 54, 1999.
- [17] U. D. Jentschura, G. Soff, and P. J. Mohr. Lamb shift of  $3p$  and  $4p$  states and the determination of  $\alpha$ . *Phys. Rev. A*, 56:1739–1755, Sep 1997.
- [18] F. Johansson. Arb: a C library for ball arithmetic. *ACM Communications in Computer Algebra*, 47(4):166–169, 2013.
- [19] F. Johansson. Rigorous high-precision computation of the hurwitz zeta function and its derivatives. *Numerical Algorithms*, 69(2):253–270, Jun 2015.
- [20] F. Johansson et al. *mpmath: a Python library for arbitrary-precision floating-point arithmetic (version 0.19)*, December 2014. <http://mpmath.org/>.
- [21] A. Laurincikas and R. Garunkstis. *The Lerch zeta-function*. Springer, 2002.
- [22] M. Lerch. Note sur la fonction  $\mathfrak{R}(w, x, s) = \sum_{k=0}^{\infty} \frac{e^{2k\pi ix}}{(w+k)^s}$ . *Acta Math.*, 11(1-4):1924, 1887.
- [23] R. Lipschitz. Untersuchung einer aus vier Elementen gebildeten Reihe. *J. Reine Angew. Math.*, 54:313–328, 1857.
- [24] G. Navas-Palencia. Fast and accurate algorithm for the generalized exponential integral  $E_{\nu}(x)$  for positive real order. *Numerical Algorithms*, 77(2):603–630, 2018.
- [25] G. Navas-Palencia. High-precision computation of the confluent hypergeometric functions via Franklin-Friedman expansion. *Advances in Computational Mathematics*, 44(3):841–859, 2018.
- [26] R. B. Paris. The Stokes phenomenon and the Lerch zeta function. *Mathematica Aeterna*, 6(2):165–179, 2016.
- [27] N. J. A. Sloane. The On-Line Encyclopedia of Integer Sequences.
- [28] J. R. Stembridge. Enriched  $p$ -partitions. *Transactions of the American Mathematical Society*, 349(2):763–788, 1997.
- [29] N. M. Temme. *Asymptotic Methods for Integrals*, volume 6 of *Series in Analysis*. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2015.

- [30] Wolfram Research, Inc. Mathematica 10.
- [31] Y. Zhuang. Counting permutations by runs. *Journal of Combinatorial Theory, Series A*, 142:147 – 176, 2016.