Springer

Dear Linas Vepstas

Here are the proofs of your article.

- You can submit your corrections **online** or by **fax**.
- For **online** submission please insert your corrections in the online correction form. Always indicate the line number to which the correction refers.
- Please return your proof together with the permission to publish confirmation.
- For **fax** submission, please ensure that your corrections are clearly legible. Use a fine black pen and write the correction in the margin, not too close to the edge of the page.
- Remember to note the journal title, article number, and your name when sending your response via e-mail, fax or regular mail.
- **Check** the metadata sheet to make sure that the header information, especially author names and the corresponding affiliations are correctly shown.
- **Check** the questions that may have arisen during copy editing and insert your answers/corrections.
- **Check** that the text is complete and that all figures, tables and their legends are included. Also check the accuracy of special characters, equations, and electronic supplementary material if applicable. If necessary refer to the *Edited manuscript*.
- The publication of inaccurate data such as dosages and units can have serious consequences. Please take particular care that all such details are correct.
- Please **do not** make changes that involve only matters of style. We have generally introduced forms that follow the journal's style.
  Substantial changes in content, e.g., new results, corrected values, title and authorship are not allowed without the approval of the responsible editor. In such a case, please contact the Editorial Office and return his/her consent together with the proof.
- If we do not receive your corrections **within 48 hours**, we will send you a reminder.

Please note

Your article will be published **Online First** approximately one week after receipt of your corrected proofs. This is the **official first publication** citable with the DOI. **Further changes are, therefore, not possible.**

After online publication, subscribers (personal/institutional) to this journal will have access to the complete article via the DOI using the URL:

`http://dx.doi.org/10.1007/s11075-007-9153-8`

If you would like to know when your article has been published online, take advantage of our free alert service. For registration and further information, go to: http://www.springerlink.com.

Due to the electronic nature of the procedure, the manuscript and the original figures will only be returned to you on special request. When you return your corrections, please inform us, if you would like to have these documents returned.

The **printed version** will follow in a forthcoming issue.

OFFPRINT ORDER
AID: **NAL79153**
MS Ref No.: **NUMA948R3**

Linas Vepstas
Austin, Texas, 78703, USA

Re: **An efficient algorithm for accelerating the convergence of oscillatory series, useful for computing the polylogarithm and Hurwitz zeta functions**

by: **Linas Vepstas**

To be published in: **Numerical Algorithms**

Dear **Linas Vepstas**

This is to let you know that the above publication has gone into production and will appear in due course. Offprints of your article may be ordered by filling out and returning this form.

I would like to receive:
0 offprints free of charge
______________ additional offprints without cover (minimum of 50 offprints)

Orders for offprints are only accepted if received with payment or if accompanied by an official purchase order from your institution, failing which no offprints can be produced. Postage and handling cost are absorbed by the publishers. Payment can be made by credit card, bankdraft personal check or international money order. UNESCO coupons are also accepted. Payment is accepted in any hard currency. Prices of additional offprints and delivery terms are mentioned on the enclosed price list.
Make checks payable to Springer Science + Business Media – Dordrecht

☐ I enclose payment to the amount of ____________________________

☐ Please charge my credit card account

Card no.: _________________________________ Expiry date: __________________________

☐ Access ☐ Eurocard ☐ American Express ☐ Bank Americard

☐ Visa ☐ Diners club ☐ Master Charge

I enclose official purchase order no.: _____________________________

VAT identification no.: ____

Date ___________________________ Signature ___________________________

PLEASE CHECK YOUR ADDRESS AND CORRECT IF NECESSARY

To: **Ruel Pinero for Springer**
E-mail: R.Pinero@spi-bpo.com
Fax: +1-703-5621913
SPi
Re: **SPi Building, Sacsac Bacong,**
Oriental Negros 6216
Philippines

Numerical Algorithms DOI 10.1007/s11075-007-9153-8
An efficient algorithm for accelerating the convergence of oscillatory series, useful for computing the polylogarithm and Hurwitz zeta functions
Vepstas

Permission to publish

I have checked the proofs of my article and

- ❑ I have **no corrections.** The article is ready to be published without changes.
- ❑ I have **a few corrections.** I am enclosing the following pages:
- ❑ I have made **many corrections.** Enclosed is the **complete article.**

Date / signature: ____

Springer
the language of science

ELECTRONIC REPRINT ORDER FORM

After publication of your journal article, electronic (PDF) reprints may be purchased by arrangement with Springer and Aries Systems Corporation.

The PDF file you will receive will be protected with a copyright system called DocuRights®. Purchasing 50 reprints will enable you to redistribute the PDF file to up to 50 computers. You may distribute your allotted number of PDFs as you wish; for example, you may send it out via e-mail or post it to your website. You will be able to print five (5) copies of your article from each one of the PDF reprints.

Please type or print carefully. Fill out each item completely.

1. Your name: ______________________________

   Your e-mail address: ______________________________

   Your phone number: ______________________________

   Your fax number: ______________________________

2. Journal title (vol, iss, pp): ______________________________
3. Article title: ______________________________
4. Article author(s): ______________________________
5. How many PDF reprints do you want? ____________________
6. Please refer to the pricing chart below to calculate the cost of your order.

Number of PDF reprints	Cost (in U.S. dollars)
50	$200
100	$275
150	$325
200	$350

NOTE: Prices shown apply only to orders submitted by individual article authors or editors. Commercial orders must be directed to the Publisher.

All orders must be prepaid. Payments must be made in one of the following forms:
- a check drawn on a U.S. bank
- an international money order
- Visa, MasterCard, or American Express (no other credit cards can be accepted)

PAYMENT (type or print carefully):

Amount of check enclosed: __________ (payable to Aries Systems Corporation)

VISA ____________________

MasterCard ____________________

American Express ____________________

Expiration date: __________ Signature: ____________________

Print and send this form with payment information to:

Aries Systems Corporation
200 Sutton Street
North Andover, Massachusetts 01845
Attn.: Electronic Reprints
— OR —
Fax this to Aries at: 978-975-3811

Your PDF reprint file will be sent to the above e-mail address. If you have any questions about your order, or if you need technical support, please contact: support@docurights.com

For subscriptions and to see all of our other products and services, visit the Springer website at:
http://www.springeronline.com

TERMS OF DELIVERY

1. A minimum of 50 offprints may be ordered. Prices corresponding to the number of pages and quantities ordered are given below.

2. Author for correspondence will receive this offprint order form. This author is therefore also responsible for any orders the co-authors wish to make. All orders for a particular paper should appear together on only one form.

3. Offprints are printed at the same time the book or journal is printed. Thus, no alterations from the exact form in which the article appears in a book or journal are possible. Any orders for offprints that are received after the book or journal is printed should be submitted to the Production Secretariat at the address below.

4. Offprints will be forwarded within a short time after the appearance of the published paper.

5. No additional offprints can be printed or sent unless this order form is filled in, signed, returned with appropriate payment or official purchase order and received before the book or journal goes to press.

6. Any correspondence in connection with offprints must state the name of the periodical, title and offprint order number and name(s) of the author(s). In case of camera-ready publications please inform us as soon as possible, preferably by fax, about the number of offprints you wish to order.
All communications should be sent to Springer, Manufacturing Department, P.O. Box 990, 3300 AZ, Dordrecht / Van Godewijckstraat 30, 3311 GX Dordrecht, The Netherlands.
ABN-AMRO Bank, Dordrecht 50.80.13.917, Postal Cheque Account Number 4447384.

Prices of additional offprints are in EUR

Number of copies	Number of pages					
	1-4	5-8	9-12	13-16	17-20	+4
50	108	177	248	318	388	71
100	153	238	323	407	492	85
150	200	300	400	500	599	100
200	246	361	476	591	706	115
250	292	422	551	681	810	130
300	339	483	628	772	917	145
350	385	544	703	863	1022	159
400	431	606	780	955	1129	175
450	478	667	857	1046	1235	189
500	524	728	932	1136	1340	204
+50	+50	+62	+72	+92	+103	+15

Payment will be accepted in any convertible currency. Please check the rate of exchange with your bank.
If ordering from within The Netherlands please add 19% VAT to the price quoted above.

As of January 1st 1993 customers within the EEC must consider the following rules:

- **If you are in possession of a VAT identification number, please fill the VAT number in on the order form. You will not be charged VAT.**
- **If you do *not* have a VAT number, then please add the low VAT rate - applicable to your country - to the prices quoted above.**

Springer
the language of science

Proof correction marks to be used by authors

Marginal mark	Meaning	Corresponding mark in text	
	Delete (take out)	/ or ⊢⊣	Cross through
	Delete and close-up		Above and below matter to be taken out
stet	Leave as printed (when matter has been crossed out by mistake)	------	Under matter to remain
caps	Change to capital letters	≡	Under letters or words altered
l.c.	Change to lower case letters		Encircle letters altered
bold	Change to bold type		Under matter altered
bold ital.	Change to bold italic type		Under matter altered
ital.	Change to italics	——	Under matter altered
rom.	Change to roman type		Encircle matter altered
×	Replace by similar but undamaged character or remove extraneous marks		Encircle letter to be altered
	Insert (or substitute) superior figure or sign	⋏ or /	
	Insert (or substitute) inferior figure or sign	⋏ or /	
	Insert (or substitute) hyphen	⋏ or /	
EN	Insert (or substitute) dash	⋏ or /	
	Insert (or substitute) solidus	⋏ or /	
•••	Insert (or substitute) ellipsis	⋏ or /	
	Close-up - delete space		Linking words or letters
#	Insert space	| or Y	Between items
equal #	Make spacing equal	|	Between items
↑	Reduce space	| or ↑	Between items
	Insert space between lines or paragraphs		
←	Reduce space between lines or paragraphs		
	Transpose		Between letters or words, numbered when necessary
	Transpose lines		
centre	Place in centre of line	] [	Around matter to be centered
	Move to the left		
	Move to the right		
NP	Begin a new paragraph	□	Before first word of new paragraph
run on	No fresh paragraph here		Between paragraphs
⋏	(Caret mark.) Insert matter indicated in margin	⋏	
	Insert single / double quotes	⋏ ⋏	

Remarks

To indicate a substitution, simply cross out the letters or words to be replaced, and write the correct letters or words in the margin. It is not necessary, nor even desirable, to use the marks for *delete* and *insert* when making a substitution. If there is more than one substitution in a line, place them in the correct order in the margin, and indicate the end of each correction with an oblique stroke / . Alternatively, continental location marks may be used, but these are to be placed in front of the corrections, not behind as in the case of the oblique stroke.
The typesetter treats *all* letters and words in the margin as insertions or substitutions, so - to avoid misunderstanding - any comments *not* intended to form part of the text should be encircled.
All alterations should be marked clearly so that there is no risk of misunderstanding; long additions or amendments should be typed on separate slips and attached. *Only really essential alterations should be made at proof stage.*
In addition to reading the proofs, please look through your edited manuscript to see if there are *any queries from the copy editor,* and if so, answer the queries *on the proofs.*

Correction Grid Form		
Line Numbers	Corrections	Remarks

Metadata of the article that will be visualized in OnlineFirst

1	Article Title	**An efficient algorithm for accelerating the convergence of oscillatory series, useful for computing the polylogarithm and Hurwitz zeta functions**	
2	Journal Name	Numerical Algorithms	
3	Corresponding Author	Family Name	**Vepštas**
4		Particle	
5		Given Name	**Linas**
6		Suffix	
7		Organization	
8		Division	
9		Address	Austin 78703, Texas, USA
10		e-mail	linasvepstas@gmail.com
11	Schedule	Received	29 May 2007
12		Revised	
13		Accepted	12 December 2007
14	Abstract	This paper sketches a technique for improving the rate of convergence of a general oscillatory sequence, and then applies this series acceleration algorithm to the polylogarithm and the Hurwitz zeta function. As such, it may be taken as an extension of the techniques given by Borwein's "An efficient algorithm for computing the Riemann zeta function" (Borwein, An efficient algorithm for the Riemann zeta function. Constructive experimental and nonlinear analysis, CMS Conference Proceedings 27, pp. 29–34 (preprint, January 1995) (2000) http://www.cecm.sfu.ca/personal/pborwein/PAPERS/P155.pdf), to more general series. The algorithm provides a rapid means of evaluating $\mathrm{Li}_s(z)$ for general values of complex *s* and a kidney-shaped region of complex *z* values given by $\|z^2/(z-1)\|<4$. By using the duplication formula and the inversion formula, the range of convergence for the polylogarithm may be extended to the entire complex *z*-plane, and so the algorithms described here allow for the evaluation of the polylogarithm for all complex *s* and *z* values. Alternatively, the Hurwitz zeta can be very rapidly evaluated by means of an Euler–Maclaurin series. The polylogarithm and the Hurwitz zeta are related, in that two evaluations of the one can be used to obtain a value of the other; thus, either algorithm can be used to evaluate either function. The Euler–Maclaurin series is a clear performance winner for the Hurwitz zeta, while the Borwein algorithm is superior for evaluating the polylogarithm in the kidney-shaped region. Both algorithms are superior to the simple Taylor's series or direct summation. The primary, concrete result of this paper is an algorithm allows the exploration of the Hurwitz zeta in the critical strip, where fast algorithms are otherwise unavailable. A discussion of the monodromy group of the polylogarithm is included.	
15	Keywords separated by ' - '	Polylogarithm - Hurwitz zeta function - Algorithm - Monodromy - Series acceleration - 65B10 (primary) - 11M35 - 11Y35 - 33F05 - 68W25	
16	Foot note information		

Numer Algor
DOI 10.1007/s11075-007-9153-8

ORIGINAL PAPER

An efficient algorithm for accelerating the convergence of oscillatory series, useful for computing the polylogarithm and Hurwitz zeta functions

Linas Vepštas

Received: 29 May 2007 / Accepted: 12 December 2007

Abstract This paper sketches a technique for improving the rate of convergence of a general oscillatory sequence, and then applies this series acceleration algorithm to the polylogarithm and the Hurwitz zeta function. As such, it may be taken as an extension of the techniques given by Borwein's "An efficient algorithm for computing the Riemann zeta function" (Borwein, An efficient algorithm for the Riemann zeta function. Constructive experimental and nonlinear analysis, CMS Conference Proceedings 27, pp. 29–34 (preprint, January 1995) (2000) http://www.cecm.sfu.ca/personal/pborwein/PAPERS/P155.pdf), to more general series. The algorithm provides a rapid means of evaluating $\mathrm{Li}_s(z)$ for general values of complex s and a kidney-shaped region of complex z values given by $\left|z^2/(z-1)\right| < 4$. By using the duplication formula and the inversion formula, the range of convergence for the polylogarithm may be extended to the entire complex z-plane, and so the algorithms described here allow for the evaluation of the polylogarithm for all complex s and z values. Alternatively, the Hurwitz zeta can be very rapidly evaluated by means of an Euler–Maclaurin series. The polylogarithm and the Hurwitz zeta are related, in that two evaluations of the one can be used to obtain a value of the other; thus, either algorithm can be used to evaluate either function. The Euler–Maclaurin series is a clear performance winner for the Hurwitz zeta, while the Borwein algorithm is superior for evaluating the polylogarithm in the kidney-shaped region. Both algorithms are superior to the simple Taylor's series or direct summation. The primary, concrete result of this paper is an algorithm allows the exploration of the Hurwitz zeta in the critical strip, where fast algorithms are otherwise unavailable. A discussion of the monodromy group of the polylogarithm is included.

L. Vepštas (✉)
Austin, Texas 78703, USA
e-mail: linasvepstas@gmail.com

Q1

Keywords Polylogarithm · Hurwitz zeta function · Algorithm · Monodromy · Series acceleration

Mathematics Subject Classifications (2000) 65B10 (primary) · 11M35 · 11Y35 · 33F05 · 68W25

1 Introduction

This article sketches a technique for improving the rate of convergence of a general oscillatory series, and then applies this technique to the computation of the polylogarithm, and to related functions, such as the Hurwitz zeta function. It essentially generalizes an algorithm given by Peter Borwein for computing the Riemann zeta function [6]. The principal result is an algorithm that efficiently obtains values of the Hurwitz zeta in the critical strip $s = \sigma + i\tau$ with $0 \leq \sigma \leq 1$.

The Hurwitz zeta function may be defined as

$$\zeta(s, q) = \sum_{n=0}^{\infty} \frac{1}{(n+q)^s} \tag{1}$$

This series is absolutely convergent for $\sigma > 1$ and any complex q that is not zero or a negative integer. There are branch points at $q = 0, -1, -2, \cdots$, rather than simple poles: this is best understood by noting that $q^s = e^{s \log q}$ and that $q = 0$ is a branch point for the logarithm, anchoring one end of a branch cut. Although the branch cut of the logarithm is conventionally taken to extend to the left, along the negative real axis, one may, in fact, run the branch cut in any direction. The Hurwitz zeta function presents a more daunting scenario: with a branch cut originating at $q = 0$ and at every negative integer, there are many different ways to navigate from sheet to sheet. Conventionally, these are all taken to lie one-atop another, along the negative real axis; this helps obscure the fact that there are a countable infinity of cuts.

The series (1) is not directly usable in the critical strip $0 \leq \sigma \leq 1$, as it does not converge. For numerical calculations, this series may be directly summed for $\sigma > 1$, although, for high-precision work, convergence becomes annoyingly slow when σ is less than about 2. The series may be analytically continued to the entire complex s-plane, excepting a simple pole at $s = 1$. The continuation, convergent on the entire complex s-plane except at the pole, was given by Helmut Hasse in 1930 [16]:

$$\zeta(s, q) = \frac{1}{s-1} \sum_{n=0}^{\infty} \frac{1}{n+1} \sum_{k=0}^{n} (-1)^k \binom{n}{k} (q+k)^{1-s} \tag{2}$$

Although the Hasse series is convergent, the convergence is far too slow for practical numerical evaluation in the critical strip.

There are other simple series that are more strongly convergent in limited domains, and so may be considered for numerical work. One such is a Taylor's series in q. To obtain this, one makes use of the derivative

$$\frac{\mathrm{d}}{\mathrm{d}q}\zeta(s,q) = -s\zeta(s+1,q)$$

to write a Taylor's series about q_0:

$$\zeta(s,q) = \sum_{n=0}^{\infty} (q_0 - q)^n \binom{s+n-1}{n} \zeta(s+n, q_0) \tag{3}$$

The binomial coefficient for a general complex-valued s can be understood to be

$$\binom{s+n-1}{n} = \frac{s(s+1)(s+2)\cdots(s+n-1)}{n!}$$

To avoid the circular problem of having to compute $\zeta(s+n, q_0)$ for some arbitrary q_0, one makes use of the fact that, for $q_0 = 1$, one has $\zeta(s,1) = \zeta(s)$, with $\zeta(s)$ being Riemann's zeta function. Since Riemann's zeta function does have several well-known algorithms [17], the Taylor's series at $q_0 = 1$ is tractable. The $q_0 = 1$ series is convergent for $|q| < 1$.

The radius of convergence for the series (3) is the least distance between q_0 and any of the branch points $q = 0, -1, -2, \cdots$. To enlarge the radius of convergence, one may explicitly handle the nearest branch-point on its own. Thus, for example, explicitly setting aside the branch point at $q = 0$ by writing $\zeta(s,q) = q^{-s} + \zeta(s, 1+q)$, one obtains

$$\zeta(s,q) = \frac{1}{q^s} + \sum_{n=0}^{\infty} (-q)^n \binom{s+n-1}{n} \zeta(s+n) \tag{4}$$

where the right-hand side employed the Taylor's expansion at $q_0 = 1$. This sum is convergent for $|q| < 1$, with the radius of convergence dictated by the branch point at $q = -1$. This process may be repeated, so that, setting aside the branch points at $q = 0, -1$ by writing

$$\zeta(s,q) = \frac{1}{q^s} + \frac{1}{(q+1)^s} + \zeta(s, 2+q)$$

and then expanding about $q_0 = 2$, one has

$$\zeta(s,q) = \frac{1}{q^s} + \frac{1}{(q+1)^s} + \sum_{n=0}^{\infty} (-q)^n \binom{s+n-1}{n} [\zeta(s+n) - 1]$$

with the sum on the right being convergent for $|q| < 2$. Repeating this process, the general form is

$$\zeta(s,q) = \sum_{k=0}^{m} \frac{1}{(k+q)^s} + \sum_{n=0}^{\infty} (-q)^n \binom{s+n-1}{n} \left[\zeta(s+n) - \sum_{k=1}^{m} \frac{1}{k^s}\right]$$

with the sum on the right being convergent for $|q| < m+1$.

Any of these expansions provide a numerically reasonable expression for the Hurwitz zeta function that is convergent on the entire complex s-plane (minus, of course, $s = 1$, and taking the appropriate limit, via l'Hôpital's rule, when s is a non-positive integer). The principal drawback to the use of these sums for high-precision calculations is the need for many high-precision evaluations of the Riemann zeta function; the computing time can grow rapidly as the number of required digits is increased.

It is the poor performance of these sums in numerical calculations that motivates the development of this paper. Since the generalized Riemann hypothesis can be phrased in terms of the values of the Hurwitz zeta function on the critical strip, it is of some interest to have a fast algorithm for computing high-precision values of this function in this region. There has been little work in this area. Ŝležavičiene provides a discussion of fast algorithms for Dirichlet L-functions in [24]. Balanzario presents an extension of the Euler–Maclaurin formula, such that it can be used to evaluate $\zeta(s, q)$ in the critical strip [3]. A particularly interesting possibility is the application of the FEE method to the evaluation of $\zeta(s, q)$ for integer s, given by Karatsuba [19].

The polylogarithm, sometimes called the *fractional polylogarithm* to emphasize that s need not be an integer, is given by the sum

$$\mathrm{Li}_s(z) = \sum_{n=1}^{\infty} \frac{z^n}{n^s} \tag{5}$$

This sum is absolutely convergent on the complex half-plane $\sigma > 0$ when $|z| < 1$, and in $\sigma > 1$ when $|z| \leq 1$. The polylogarithm has a non-trivial analytic structure. For fixed s, the principal sheet has a branch point at $z = 1$, with a branch cut conventionally taken on the positive real axis, for $1 < z < +\infty$. The location of the branch-point at $z = 1$, as is always the case with branch points, is the cause of limited convergence of the analytic series in the complex z-plane. The other sheets of the polylogarithm also have a branch point at $z = 0$, which is not naively obvious from the form (5). The set of possible paths that navigate from sheet to sheet form a group, known as the *monodromy group* of the function; a detailed exposition of the monodromy will be given in this paper.

The Hurwitz zeta may be expressed in terms of the polylogarithm [20] by means of Jonquière's identity [18, 20, Section 7.12.2]

$$\mathrm{Li}_s\left(e^{2\pi i q}\right) + (-1)^s\, \mathrm{Li}_s\left(e^{-2\pi i q}\right) = \frac{(2\pi i)^s}{\Gamma(s)} \zeta(1-s, q) \tag{6}$$

which holds for the region $\sigma > 1$, $0 \leq \Re q < 1$, and $\Im q > 0$. Thus, fast algorithms for the polylogarithm can be converted to fast algorithms for the Hurwitz zeta. Wood [30] provides a extensive review of the means of computing the polylogarithm, but limits himself to real values of s. Crandall [10] discusses evaluating the polylogarithm on the entire complex z-plane, but limits himself to integer values of s. Thus, there appears to be a similar paucity of general algorithms for the polylogarithm as well. The series defining the polylogarithm may be directly evaluated when $|z| < 1$, although direct evaluation becomes quite slow as $|z|$ approaches 1 and σ is smaller than about 2. There do not seem to be any published algorithms that may be applied on the critical strip.

The primary effort of this paper is to take the algorithm given by Borwein [6], which is a simple Padé-type approximant algorithm [7], and generalize it to the polylogarithm. The result is a relatively small finite sum that approximates the polylogarithm, and whose error, or difference from the exact value, can be precisely characterized and bounded. Increased precision is easily obtainable by evaluating a slightly larger sum; one may obtain roughly $2N$ to $4N$ bits of precision by evaluating a sum of N terms. The sum may be readily evaluated for just about any value s on the complex s-plane. However, it is convergent only in a limited region of the z-plane, and specifically, is only convergent when

$$\left|\frac{z^2}{z-1}\right| < 4$$

This is sufficient for evaluating the Hurwitz zeta for general complex s and real $0 < q < 1$. Unfortunately, there does not appear to be any simple and efficient way of extending this result to the general complex z-plane, at least not for general values of s. By using duplication and reflection formulas, one may extend the algorithm to the entire complex z-plane; however, this requires the independent evaluation of Hurwitz zeta.

The extended Borwein algorithm described here can be considered to be a yet another form of series acceleration, applied in a particular context. As there already exists a large body of work on series acceleration [8, 14, 27], a few words are in order for the choice made here. First, much of the existing literature is focused on real-valued series. When a formal series, such as $\sum_n a_n z^n$ is considered, it is usually implicitly assumed that z is real, and positive. Alternately, some of the finest and most powerful series acceleration algorithms take $z = -1$, so that the sequence being summed is a real-valued, alternating sequence. By contrast, the problem at hand takes z to be complex-valued, and so, in this sense, *oscillatory*, as $z = re^{i\theta}$ with θ non-zero. In addition, the a_n show a weak, complex-valued oscillation, with the $|a_n|$ being only logarithmically convergent. Thus, most of the well-known series acceleration techniques do not appear to be *prima facie* applicable to the problem at hand. As to numerical performance, the best techniques, and in particular, the non-linear techniques, seem to offer approximately $2N$ to $4N$ bits of precision for a logarithmically convergent series of N terms, which is comparable to the results obtained here. This is supported by Jentschura et al. [25], who apply several different non-linear sequence transformations to the Riemann zeta function and the Lerch transcendent. Importantly, the Borwein algorithm provides an exact, explicit error bound on the accuracy of results. This is particularly important for implementation in general precision algorithm libraries, where error and rounding control are vital for accurate results. Whether any of the other well-known techniques can be adapted to this problem, whether they offer equivalent or superior numerical space/time performance, and whether they offer explicit error estimation, remains an open question.

Although the Hurwitz zeta function may be computed by evaluating the Taylor's series (4) directly, a superior approach is to perform an Euler–Maclaurin summation (thanks to Oleksandr Pavlyk for pointing this out [Pavlyk, private communication, Wolfram Research, February 2007]). The summation uses the standard, textbook Euler–Maclaurin formula [1, 25, 4, 7], and is applied to the function $f(x) = (x+q)^{-s}$. However, the application is "backwards" from the usual sense: the integral of $f(x)$ is known (it is easily evaluated analytically), and it is the series, which gives the Hurwitz zeta, which is unknown. All of the other parts of Euler–Maclaurin formula are easily evaluated. The result is an algorithm that is particularly rapid for the Hurwitz zeta function. This, and some of the other algorithms discussed above, have been implemented, and a measurement of their relative performance was taken. This data is reviewed in a later section. It indicates that Euler–Maclaurin summation outperforms the direct evaluation of the Taylor's series by orders of magnitude. It also indicates that Euler–Maclaurin summation is faster than evaluating the polylogarithm twice, and using Jonquière's identity (6).

The development of this paper proceeds as follows. The next section presents a general method for summing oscillatory sequences. This is followed by a section containing a few lemmas needed to support the general theory. These are then applied to a certain integral representation of the polylogarithm. A polynomial series for the polylogarithm is given, and the error term of the resulting approximation is precisely bounded. This is followed by a very short review of the application of Euler–Maclaurin summation. There is a brief review of the duplication formula for the Hurwitz zeta, and a short discussion of ways in which this numerical algorithm may be tested for correctness. Measurements of the performance of actual implementations of the various numerical algorithms is provided. The paper concludes with a detailed derivation of the monodromy group, and a discussion of Apostol's "periodic zeta function".

The algorithm has been implemented by the author using the Gnu Multiple Precision arithmetic library [13], and is available on request, under the terms of the LGPL license. The paper concludes with some intriguing images of the polylogarithm and the Hurwitz zeta function on the critical strip.

2 Convergence acceleration of oscillatory series

This section considers series acceleration techniques for the formal series

$$S(z) = \sum_{n=0}^{\infty} a_n z^n \tag{7}$$

where z is understood to be complex-valued and a priori unrestricted. Informally, this series will be termed *oscillatory*, in the sense that when z lies on the unit circle $|z| = 1$, the powers z^n are seen to circle around the origin. The goal is to present techniques that converge more rapidly than the simple partial sums of (7).

One of the oldest techniques for the acceleration of convergence is Euler's transformation of an alternating series [1, Eqn. 3.6.27]. An alternating series can be considered to be a special case of (7), with z set to $z = -1$. Euler's transformation is most clearly expressed by making use of the forward difference operator. This operator, $\triangle$, is defined to act on a sequence $\{a_n\}$ as $\triangle a_n = a_{n+1} - a_n$. By abuse of notation, it acts on a general function $f(x)$ as $\triangle f(x) = f(x+1) - f(x)$. Repeated iteration is used to define $\triangle^n$, so that $\triangle^2 f(x) = f(x+2) - 2f(x+1) + f(x)$ and

$$\triangle^n f(x) = \sum_{k=0}^{n} (-1)^{n-k} \binom{n}{k} f(x+k) \tag{8}$$

Euler then considers a convergent alternating series

$$S = \sum_{n=0}^{\infty} (-1)^n a_n$$

and finds that it can be re-written as

$$S = \sum_{n=0}^{\infty} \frac{(-1)^n \triangle^n a_0}{2^{n+1}} \tag{9}$$

Aside from the power of two in the denominator aiding in convergence, it is often the case that the forward differences themselves rapidly become small. Euler's method effective, although not very strong, and it has long been employed numeric calculations. A particularly efficient algorithm for Euler's method is given by van Wijngaarden [11, 29].

Euler's transformation of series has several general forms. One particularly enlightening form [22, page 311] is

$$\sum_{n=0}^{\infty} \frac{a_n}{t^{n+1}} = \sum_{k=0}^{\infty} \frac{\triangle^k a_0}{(t-1)^{k+1}} \tag{10}$$

where the right hand side converges for $|t-1| > R+1$ when the left hand side is convergent for $|t| > R$. Euler's transform for alternating series is regained by setting $t = -1$. Setting $z = 1/t$, the above takes a form that suggests the later developments in this section:

$$\sum_{n=0}^{\infty} a_n z^n = \frac{1}{1-z} \sum_{k=0}^{\infty} \left(\frac{z}{1-z}\right)^k \triangle^k a_0 \tag{11}$$

A good example of the strength (and weakness) of Euler's transformation is its application to the Hurwitz zeta function, where it essentially leads to the

Hasse series (2). Taking $f(x)$ to be q^{1-s} in (8), the inner sum of the Hasse series (2) is then

$$\sum_{k=0}^{n}(-1)^{n-k}\binom{n}{k}(q+k)^{1-s} = \triangle^n q^{1-s}$$

The benefit of the Hasse series is that it has improved convergence to the point of being globally convergent on the complex s-plane, whereas the traditional series representation of (1) is not. Yet, things can be improved further still. A quick numerical experiment will show that the Hasse series does not converge rapidly, especially in the critical strip. The attempt to improve convergence even further, beyond this traditional technique, motivates the development of this paper.

Strictly speaking, the slow convergence can be attributed to the fact that the naive series representations (1) or (5) are not strictly alternating series, as would be properly required to justify the application of the Euler method. Instead, the series are a superposition of oscillatory terms; for the polylogarithm, an explicit oscillation coming from $z^n = |z|^n e^{in \arg z}$ and a slower oscillation coming from $n^{-s} = n^{-\sigma} e^{-i\tau \log n}$. Thus, one would like to have a theory of sequence acceleration not for alternating series, but for oscillatory series.

One problem with the (11) is that, for $|z| = 1$, it is optimal only when $z=-1$, and becomes a de-acceleration technique as z approaches 1. One would like to somehow "twist" this series, so that it is optimally convergent for any particular z. This may be done by replacing the forward difference operator $\triangle a_n = a_{n+1} - a_n$ by $\triangle_q a_n = a_{n+1} - q a_n$. Straightforward manipulations lead to

$$\frac{1}{1-qz}\sum_{m=0}^{\infty}\left(\frac{z}{1-qz}\right)^m \triangle_q^m a_0 = \sum_{n=0}^{\infty} z^n a_n$$

with the traditional Euler's series (11) regained by taking $q = 1$. In this form, the series is reminiscent of the starting point for Padé-type approximations [7]. Assuming that the oscillatory part z is known, or can be approximately guessed at from the period of the oscillation, then choosing $q = -1/z$ should make the left hand side an accelerated series for the right hand side.

There is a more powerful acceleration technique; it is at the center of this paper and is described now. Suppose that it is possible to express the a_n of (7) as moments, so that

$$a_n = \int_0^1 y^n g(y) \mathrm{d}y$$

for some function $g(y)$. Since $g(y)$ is still fairly general (subject to constraints discussed below), this assumption does not overly restrict the a_n, and should not bar most practical applications. It follows that

$$S(z) = \int_0^1 \frac{g(y)}{1-yz} \mathrm{d}y \tag{12}$$

The goal of the series acceleration is to write

$$S(z) = S_n(z) + \varepsilon_n(z)$$

in such a way that $S_n(z)$ can be computed rapidly, and $\varepsilon_n(z)$ is a small and explicitly bounded error term. This is done in two steps, known to this author by generalizing from Borwein [6] and Cohen et al. [26], although the technique is possibly older.

First, one makes an Ansatz and writes

$$S_n(z) = \frac{1}{p_n\left(\frac{1}{z}\right)} \int_0^1 \frac{p_n\left(\frac{1}{z}\right) - p_n(y)}{1 - yz} g(y)\, \mathrm{d}y \tag{13}$$

and

$$\varepsilon_n(z) = \frac{1}{p_n\left(\frac{1}{z}\right)} \int_0^1 \frac{p_n(y) g(y)}{1 - yz}\, \mathrm{d}y$$

so that $S(z) = S_n(z) + \varepsilon_n(z)$ holds vacuously, by rearrangement of terms. The second step is to find a sequence of polynomials $\{p_n(y)\}$ such that the $S_n(z)$ are easily evaluated, while also showing that the $\varepsilon_n(z)$ are bounded and geometrically decreasing. The next section will show that $S_n(z)$ can be written as

$$S_n(z) = \frac{-1}{p_n\left(\frac{1}{z}\right)} \sum_{k=0}^{n-1} c_k a_k \tag{14}$$

With a suitable choice of $p_n(z)$, the coefficients c_k are presumably easy to evaluate. To show that the $S_n(z)$ is really a series acceleration for the formal series (7), one must show that $|\varepsilon_n(z)|$ is bounded. If $g(y) \geq 0$, then

$$|\varepsilon_n(z)| \leq \left| \frac{p_n(y_0)}{p_n\left(\frac{1}{z}\right)} \right| |S(z)|$$

where y_0 is the point on the unit interval where $p_n(y)$ assumes its maximum. Provided that one can find a polynomial sequence such that

$$\left| \frac{p_n(y_0)}{p_n\left(\frac{1}{z}\right)} \right| \sim A^{-n}$$

for some real number $A > |1/z|$, then one has that the series $S_n(z)$ converges to $S(z)$ more rapidly than the simple partial sums $\sum_{k=0}^{n} z^k a_k$ of (7).

The above is a demonstration of a series acceleration method for a more-or-less general sequence of a_n, and is in no way particular to the polylogarithm or the Hurwitz zeta. The only ingredients of the demonstration were that the a_n are "well-behaved" – in this case, being expressible as moments of a general function $g(y)$. For the conclusion to hold, $g(y)$ needs to be integrable; it need not be analytic, differentiable or even continuous. Thus, one might hope that the acceleration method might work, even for those general cases where $g(y)$ is not explicitly known, but the a_n are somehow "reasonable".

For the special case of $z = -1$, i.e. for the case of an alternating series, Cohen et al. [26] suggest some remarkably strongly converging polynomials $p_n(y)$, with values of A from 5.8 to as much as 17.9. The generalization of those results to arbitrary z is not immediately apparent, but is surely possible. By comparison, Borwein's algorithm 2 [6], using the Tchebysheff polynomials, corresponds to $A = 3 + \sqrt{8} \approx 5.828$ and algorithm 3, generalized in a later section, has $A = 8$.

3 Polynomial sequences

This section provides a series of short lemmas to establish the result (14).

Lemma 3.1 *Given a polynomial $p_n(y)$ of degree n, it can be shown that*

$$r_n(y) = \frac{p_n(y) - p_n(1/z)}{1 - yz} \tag{15}$$

is again a polynomial in y, of degree $n-1$, for any constant z, provided that $z \neq 0$.

Proof This may be easily proved, term by term, by noting that $(x^n - a^n)/(x - a)$, with $a = 1/z$, has the desired properties.

An explicit expression for $r_n(y)$ is needed. Write the polynomial $p_n(y)$ as

$$p_n(y) = \sum_{k=0}^{n} b_k y^k \tag{16}$$

while, for r_n, assume only a general series:

$$r_n(y) = \sum_{k=0}^{\infty} c_k y^k \tag{17}$$

Setting $y = 0$, one immediately obtains $c_0 = b_0 - p_n(1/z)$. Higher coefficients are obtained by equating derivatives:

$$r_n^{(k)}(y) = \frac{1}{1 - yz}\left[p_n^{(k)}(y) + kzr_n^{(k-1)}(y)\right]$$

where $r_n^{(k)}(y)$ is the k'th derivative of r_n with respect to y. Setting $y = 0$ in the above, one obtains the recurrence relation $c_k = b_k + zc_{k-1}$ which is trivially solvable as

$$c_k = z^k \left[-p_n\left(\frac{1}{z}\right) + \sum_{j=0}^{k} \frac{b_j}{z^j}\right] \tag{18}$$

From this, it is easily seen that $c_n = 0$ and more precisely $c_{n+m} = 0$ for all $m \geq 0$. This confirms the claim that $r_n(y)$ is a polynomial of degree $n - 1$ in y.

To prove (14), one notes simply that (15) is the integrand to (13). Then, substituting (17) for the integrand, keeping in mind that this last series terminates at $n-1$, one finally arrives at (14).

4 The polylogarithm

The development of the algorithm requires the following integral representation for the polylogarithm:

$$\mathrm{Li}_s(z) = \frac{z}{\Gamma(s)} \int_0^1 \frac{|\log y|^{s-1}}{1-yz}\, \mathrm{d}y \tag{19}$$

This is readily obtained by using the integral representation for $\Gamma(s)$ to replace n^{-s} in the series representation (5) for $\mathrm{Li}_s(z)$, and then exchanging the order of summation and integration. A comprehensive development of this integral representation is given by Costin and Garoufalidis [9, Thm. 1].

This integral representation is precisely of the form of the quadrature (12). Following the development above, one has

$$\begin{aligned} \varepsilon(s,z) &= \frac{1}{f(1/z)} \frac{z}{\Gamma(s)} \int_0^1 \frac{f(y)\,|\log y|^{s-1}}{1-yz}\, \mathrm{d}y \\ &= \mathrm{Li}_s(z) + \frac{1}{f(1/z)} \frac{z}{\Gamma(s)} \int_0^1 \frac{f(y)-f(1/z)}{1-yz}\, |\log y|^{s-1}\, \mathrm{d}y \end{aligned} \tag{20}$$

Replacing $f(y)$ with a polynomial $p_n(y)$, and making use of the explicit form (18), one obtains

$$\varepsilon_n(s,z) = \mathrm{Li}_s(z) + \frac{z}{p_n(1/z)} \sum_{k=0}^{n-1} \frac{c_k}{(k+1)^s} \tag{21}$$

Thus, to obtain a good approximation for $\mathrm{Li}_s(z)$, one needs to find a polynomial sequence such that ε_n goes to zero as $n \to \infty$ for the domain (s,z) of interest. That is, one seeks to make

$$\varepsilon_n(s,z) = \frac{1}{p_n(1/z)} \frac{z}{\Gamma(s)} \int_0^1 \frac{p_n(y)\,|\log y|^{s-1}}{1-yz}\, \mathrm{d}y \tag{22}$$

as small as possible. Many different polynomial sequences can be contemplated. They need to have two properties: they should minimize the integrand, while also maximizing the value of $|p_n(1/z)|$. There is some advantage to keeping these simple, so that the $c_k$are easily computed. One possible sequence, based on the Bernoulli process or Gaussian distribution, is explored in the next section.

5 The Gaussian distribution

In order to suppress the logarithmic branch point at $y = 0$ in the integrand of (22), one wishes the small y behavior of $p_n(y)$ to be $p_n(y) \sim y^a$ for $a > \Re s - 1$. In this section, the sequence $p_{2n}(y) = y^n(1-y)^n$ is considered. As is well known from probability theory, where y can be interpreted as the probability of flipping *heads* on a fair coin toss (a *Bernoulli process*), this sequence goes over to the Gaussian distribution for large n. It is useful here because it becomes sharply peaked, subtending a diminishingly small area, thus minimizing the integrand. For large values of n, it is peaked at $y = 1/2$, falling off rapidly away from that point, as it strongly approximates a Gaussian:

$$p_{2n}(y) = \frac{1}{4^n} e^{-4n(y-1/2)^2} \left[1 + 2\left(y - \frac{1}{2}\right)^2 + \mathcal{O}\left(\left(y - \frac{1}{2}\right)^4\right)\right]$$

Here, the order-of notation $\mathcal{O}$ simply signifies the order of the corrections as one moves away from the peak at $y = 1/2$. From basic probability theory, it is known that this approximation gets increasingly good as n increases.

Using this in the integral (22), and assuming that the real part of z is not positive, it is not hard to deduce the very crude estimate

$$|\varepsilon_n(s,z)| \sim \left| \left(\frac{z^2}{4(z-1)}\right)^n \frac{z}{\Gamma(s)} \right|$$

which confirms that $\varepsilon_n(s,z)$ does indeed get suitably small for a certain region in the complex z-plane. However, in order for (21) to be useful computationally, an upper bound on the value of ε_n needs to be given, as a function of z and n. This bound is derived in the next section.

The polynomial coefficients of (16) are easily obtained, and are

$$b_0 = b_1 = \cdots = b_{n-1} = 0$$

$$b_{n+k} = (-1)^k \binom{n}{k} \qquad \text{for } 0 \le k \le n$$

This leads to

$$\mathrm{Li}_s(z) = -\frac{z^{2n+1}}{(z-1)^n} \sum_{k=0}^{2n-1} \frac{c_k}{(k+1)^s} + \varepsilon(s,z)$$

where the c_k, the same as in (18), are given by

$$c_k = z^k \left[-\left(\frac{z-1}{z^2}\right)^n + \frac{1}{z^n} \sum_{j=0}^{k-n} (-1)^j \binom{n}{j} \frac{1}{z^j} \right]$$

The summation above is to be understood to be zero when $k < n$. The above summations can be re-organized into the more suggestive form

$$\mathrm{Li}_s(z) = \sum_{k=1}^{n} \frac{z^k}{k^s} + \frac{1}{(1-z)^n} \sum_{k=n+1}^{2n} \frac{z^k}{k^s} \sum_{j=0}^{2n-k} (-z)^j \binom{n}{j} + \varepsilon_n(s,z) \tag{23}$$

The error term $\varepsilon_n(s, z)$ is negligible for only a very specific area of the complex z-plane. The region where ε_n may be ignored as a small error is derived in the next section. Substituting $z = -1$ in the above formulas agrees with expressions previously given by Borwein [6].

The above formula has been implemented using the Gnu Multiple Precision arithmetic library (GMP) [13], and has been numerically validated for correctness in several different ways. The source code, under the license terms of the LGPL, may be obtained by contacting the author.

6 Bound on the error term

In order for (23) to be useful computationally, an upper bound on the value of $|\varepsilon_n|$ as a function of n needs to be given. To compute the polylogarithm to some desired precision, one infers a suitable value of n based on this bound. However, to achieve this desired precision, one must not only choose n large enough, but one must also maintain a somewhat larger number of digits in the intermediate terms, as the appearance of the binomial coefficient in the (23) implies that intermediate terms can become quite large, even while mostly canceling. This section derives an upper bound on the size of ε_n, for the Gaussian distribution, and briefly discusses the issue of the required precision in intermediate terms.

The general behavior of the integrand appearing in (22) depends on the whether $\Re s \geq 1$ or not; estimates are presented for these two cases. Writing $s = \sigma + i\tau$ for real σ and τ, and assuming $\sigma \geq 1$ and choosing n so that $\sigma \leq n$, one has

$$\begin{aligned} \left| p_{2n}(y) \left|\log y\right|^{s-1} \right| &\leq p_{2n}(y) \left|\log y\right|^{\sigma-1} \\ &= \left|y(1-y)\log y\right|^{\sigma-1} \left(y(1-y)\right)^{n-\sigma+1} \end{aligned}$$

Each part of the right hand side is bounded by a Gaussian, and since the product of Gaussians is a Gaussian, so is the entire expression. From the Stirling approximation, one has the well-known identity

$$|y(1-y)|^a \leq \frac{1}{4^a} \exp -4a \left(y - \frac{1}{2}\right)^2$$

The other part is also bounded by a Gaussian

$$|y(1-y)\log y|^a \leq |y_0\,(1-y_0)\log y_0|^a \exp\left(-\frac{a\,(y-y_0)^2}{2y_0\,(1-2y_0)}\right)$$

which is centered on the maximum of $|y_0\,(1-y_0)\log y_0|$, that is, at $y_0 = 0.23561058253\cdots$, where y_0 is the solution to

$$0 = \frac{d}{dy}\, y(1-y)\,|\log y|$$

Curiously, the numerical value of the factor in the Gaussian is $1/2y_0(1-2y_0) = 4.013295587\cdots$

To evaluate the integral in (22), one also needs a bound on the denominator. This is furnished by

$$\left|\frac{1}{1-yz}\right| \le C(z) \equiv \begin{cases} 1 & \text{if } \Re z \le 0 \\ \frac{1}{|1-z|} & \text{if } 0 < \Re z \ \text{ and } \ |z| < 1 \\ \left|\frac{z}{\Im z}\right| & \text{if } 0 < \Re z \ \text{ and } \ |z| > 1 \end{cases}$$

Thus, the integrand is bounded by a Gaussian. Multiplying the two Gaussians, completing the square, and evaluating the resulting integral is a bit tedious. The result is that

$$\left|\int_0^1 \frac{p_{2n}(y)\,|\log y|^{s-1}}{1-yz}\,dy\right| \le C(z)\frac{|4y_0\,(1-y_0)\log y_0|^{\sigma-1}}{4^n}G(s)$$

The constant may be taken as $|4y_0\,(1-y_0)\log y_0| = 1.041381965\ldots$. The factor $G(s)$ containing the completed square of the Gaussians is bounded by

$$\begin{aligned} G(s) &\le \sqrt{\frac{\pi}{4n}}\exp -(1-2y_0)^2\,(\sigma-1)\,\frac{8y_0\,(1-2y_0)}{8y_0\,(1-2y_0)+\frac{\sigma-1}{n-\sigma+1}} \\ &\le \sqrt{\frac{\pi}{4n}}\exp -(1-2y_0)^2\,\frac{\sigma-1}{\sigma} \\ &\le 1 \end{aligned}$$

when $1 \le \sigma$. Useful for estimation is that $8y_0(1-2y_0) = 0.996687115\ldots$. Combining these results, one obtains

$$|\varepsilon(s,z)| \le \frac{1}{4^n}\left|\frac{z^2}{z-1}\right|^n\left|\frac{z}{\Gamma(s)}\right| C(z)G(s)\,(1.0414\ldots)^{\sigma-1} \tag{24}$$

for the case where $1 \le \sigma$ and n chosen so that $\sigma \le n$.

The case where $\sigma \le 1$ may be evaluated as follows. One writes

$$\begin{aligned} \left|p_{2n}(y)\,|\log y|^{s-1}\right| &\le p_{2n}(y)\,|\log y|^{\sigma-1} \\ &= \left|\frac{1-y}{\log y}\right|^{1-\sigma} y^n(1-y)^{n+\sigma-1} \\ &\le y^n(1-y)^{n+\sigma-1} \end{aligned}$$

The integrand may now be recognized as the beta function, so that

$$\left|\int_0^1 \frac{p_{2n}(y)\,|\log y|^{s-1}}{1-yz}\,dy\right| \le C(z)\frac{\Gamma(n+1)\Gamma(n+\sigma)}{\Gamma(2n+\sigma+1)}$$

In the region where $|\sigma| \ll n$, one may approximate

$$\frac{\Gamma(n+1)\Gamma(n+\sigma)}{\Gamma(2n+\sigma+1)} \le \frac{2^{1-\sigma}}{4^n}$$

The above follows readily from the Stirling approximation

$$\Gamma(x) \sim \sqrt{\frac{2\pi}{x}} \left(\frac{x}{e}\right)^x$$

which is the first term of a well-known asymptotic expansion [1, Eqn. 6.1.37] for the gamma function.

Combining the various pieces, one obtains a result remarkably similar to the previous one; namely, when $\sigma \leq 1$ but n is such that $|\sigma| \ll n$, one has

$$|\varepsilon(s, z)| \leq \frac{1}{4^n} \left|\frac{z^2}{z-1}\right|^n \left|\frac{z}{\Gamma(s)}\right| 2^{1-\sigma} C(z) \tag{25}$$

For evaluations on the critical line $\sigma = 1/2$, one needs a good estimate for $|\Gamma(s)|^{-1}$, which can become quite large as the imaginary part of s increases. A useful estimate for $|\Gamma(s)|^{-1}$ is given in [6], for the case where $\sigma \geq -m + 1/2$:

$$\begin{aligned} \frac{1}{|\Gamma(s)|} &= \frac{1}{|\Gamma(\sigma)|} \sqrt{\prod_{k=0}^{\infty} \left(1 + \frac{\tau^2}{(\sigma+k)^2}\right)} \\ &\leq \frac{1}{|\Gamma(\sigma)|} \sqrt{\frac{\sinh \tau\pi}{\tau\pi} \prod_{k=0}^{m} \left(1 + \frac{\tau^2}{\left(\frac{1}{2}+k\right)^2}\right)} \end{aligned}$$

The first equality is known from [1, Eqn. 6.1.25], while the inequality follows from the theory of elementary transcendental functions, and [1, Eqn. 4.5.68].

A remarkable side-effect of the estimation is that, for s equal to negative integers, one has that $\Gamma(s)$ is infinite, thus implying that the error term is zero. This somewhat surprising result corresponds to the fact that $\mathrm{Li}_{-n}(z)$ has an exact expression as a ratio of two polynomials, the denominator being of degree $n+1$. Indeed, setting $s = 0$ and so $n = 1$ into (23) gives the exact result

$$\mathrm{Li}_0(z) = \frac{z}{1-z}$$

while for $s = -1$, one must use $n = 2$, to obtain the exact result

$$\mathrm{Li}_{-1}(z) = \frac{z}{(1-z)^2}$$

The generic form for the polylogarithm at the negative integers [20] is

$$\mathrm{Li}_{-n}(z) = (-1)^n \sum_{k=0}^{n} (-1)^k \frac{k!}{(1-z)^{k+1}} \left\{ \begin{matrix} n+1 \\ k+1 \end{matrix} \right\}$$

where $\left\{ \begin{matrix} n \\ k \end{matrix} \right\}$ denotes the Stirling numbers of the second kind, which are given by the k'th forward difference of x^n evaluated at $x = 0$:

$$\left\{ \begin{matrix} n \\ k \end{matrix} \right\} = \left.\frac{\Delta^k x^n}{k!}\right|_{x=0} = \frac{1}{k!} \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} j^n$$

In general, the formula (23) seems to have the curious property of always evaluating to exactly the same rational function whenever s is a non-positive integer, and $-s < n$. That is, for fixed non-positive integer s, the sum is independent of n, provided that n is large enough. Thus, this polynomial approximation has the pleasing result of giving exact answers in the case when an exact polynomial answer is possible.

For the general case, one may conclude that the estimate (23) can be effectively applied whenever

$$\left|\frac{z^2}{z-1}\right| < 4$$

To obtain a value of the polylogarithm to within some given numerical precision, one must invert the formulas (24) or (25), solving for the value of n which is to be used in (23). To obtain a fixed number of digits of precision, one must carry out intermediate calculations with at least that many digits of precision. In fact, one must have more. The appearance of the binomial coefficient in (23) is the problem. Since $2^n > \binom{n}{j}$, one concludes that n additional binary bits of precision are needed with which to carry out the calculations. As a result, it becomes difficult to implement the algorithm robustly using only double-precision arithmetic. For values of z near $z = -1$, the algorithm is well-enough behaved; however, round-off errors significantly disturb the calculations as z approaches $+1$. This trouble with precision can be at least partly alleviated by making use of the duplication formula, as discussed in the next section.

7 Euler–Maclaurin summation

An alternative algorithm for computing the Hurwitz zeta may be based upon the Euler–Maclaurin series. This algorithm proves to be quite rapid and efficient (Pavlyk, private communication, Wolfram Research, February 2007). A detailed discussion of the application of the Euler–Maclaurin formula to the Riemann zeta function is given by Weniger and Kirtman [12, Section 2]. In particular, the methods discussed by Weniger in [28] should be directly extendable to the Hurwitz zeta function. This section provides only a basic sketch.

The Euler–Maclaurin series may be written as [1, Eqn. 23.12.30]

$$\sum_{k=0}^{\infty} f(k) = \sum_{k=0}^{N-1} f(k) + \frac{f(N)}{2} + \int_N^\infty f(x)\mathrm{d}x - \sum_{k=1}^{p} \frac{B_{2k}}{(2k)!} \frac{d^{2k+1}}{\mathrm{d}x^{2k+1}} f(x)\Big|_{x=N} + R \tag{26}$$

This formula is simply applied to the function $f(x) = (x+q)^{-s}$ to obtain the Hurwitz zeta. The derivative is particularly easy to evaluate:

$$\frac{\mathrm{d}^{2k+1}}{\mathrm{d}x^{2k+1}} \frac{1}{(x+q)^s} = -\frac{s(s+1)(s+2)\cdots(s+2k)}{(x+q)^{s+2k+1}}$$

and the integral is trivial:

$$\int_N^\infty \frac{1}{(x+q)^s}\,\mathrm{d}x = \frac{1}{(s-1)}\,\frac{1}{(N+q)^{s-1}}$$

The error made by this expansion is embodied in the term R. It is directly given by [1, Eqn. 23.1.32]

$$R = \frac{2}{(2\pi)^{2p}} \int_N^\infty f^{(p+1)}(x) \frac{B_{p+1}\left(x - \lfloor x \rfloor\right)}{(p+1)!}\mathrm{d}x$$

where the $B_k(x)$ are the Bernoulli polynomials [1, chap. 23.1]. These are defined by the generating function

$$\frac{te^{xt}}{e^t - 1} = \sum_{n=0}^{\infty} B_n(x) \frac{t^n}{n!}$$

The notation $\lfloor x \rfloor$ denotes the floor of x, that is, the largest integer less than x.

There are various formulas in the literature which may be readily applied to bound the size of this term [1]. Thus, for example, one has

$$R \leq \frac{2}{(2\pi)^{2p}} \frac{|s(s+1)\cdots(s+2p)|}{\Re s + 2p} \frac{1}{(N+q)^{\Re s + 2p}}$$

Thus, the evaluation of the sum (26) merely requires a suitable choice of N and p to be used. As it happens, the situation is even simpler than this. The sum (26) is an asymptotic series, and it is well-known that the best approximation for an asymptotic series occurs when one stops the summation at the smallest term in the series. Thus, it is sufficient to choose a value of N, while p is found dynamically as the algorithm runs.

There remains the question of what value of N to use. This remains an open problem, which will not be explored here. However, it seems that an adequate heuristic for most common cases is to choose $N = D/2 + 10$ where D is the desired number of decimal digits of precision. The performance of this algorithm is compared to that of the Borwein algorithm in the next section.

8 Performance

The performance of both Euler–Maclaurin summation and the Borwein algorithm appears to be very good. As a general rule, both have better performance than the direct summation formula (5), although this depends on whether the polylogarithm or the Hurwitz zeta is the desired quantity.

A proper performance analysis of an algorithm would provide a detailed accounting of the number of multiplications required to implement the algorithm. This would allow the comparison of different algorithms for optimality. Such a detailed accounting is beyond the scope of this paper. In part this is because the algorithms require the exponentiation of complex numbers, the calculation of Bernoulli numbers and binomial coefficients, and, if Jonquière's identity (6) is used, the evaluation of $\Gamma(s)$ for complex s. The performance evaluation of each of these steps presents its own difficulties. In addition, various sub-expressions can be cached, if it is known that either s or q or z will be held fixed from one evaluation to the next. Additional performance improvements can be obtained if any of the parameters are real-valued instead

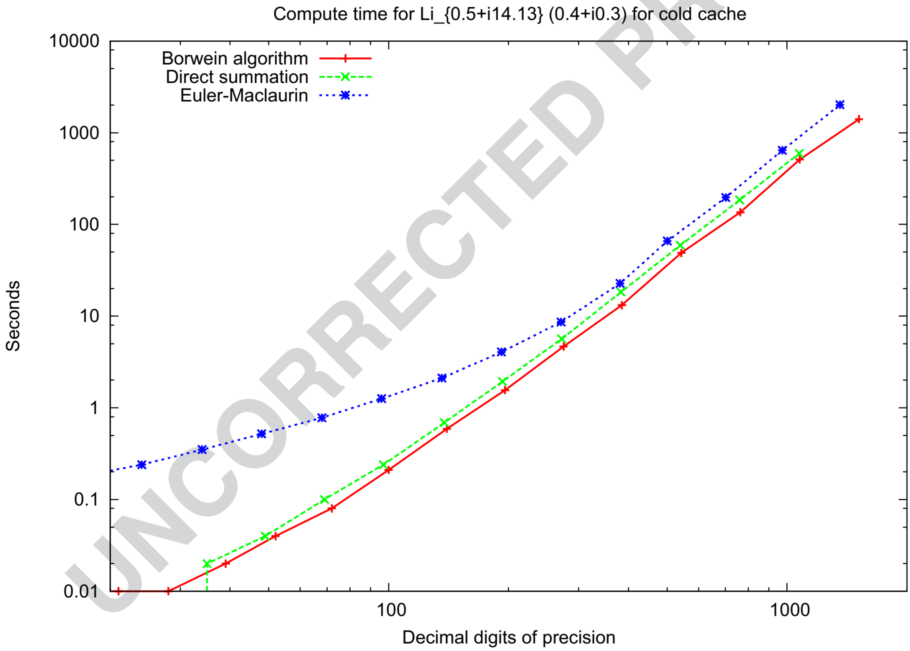

Fig. 1 Polylogarithm cold cache performance. This figure shows the time, in seconds, needed to compute the polylogarithm to the indicated number of decimal places of precision. The values chosen for evaluation are $s = 0.5 + i14.134725$, which is very near to the first of the non-trivial Riemann zeroes, and $z = 0.4 + i0.3$, which is relatively near to the origin. For such a small value of z ($|z| = 0.5$), one might hope that direct summation might compete favorably against the other two algorithms; and in particular against the Borwein algorithm, as the point is closer to the troublesome branch point at $z = +1$ than algorithmically optimal argument $z = -1$. Nevertheless, $\left|z^2/(z-1)\right| \approx 0.373$ for this point, and the Borwein algorithm wins. The Euler–Maclaurin algorithm is relatively disadvantaged, as it needs to be evaluated twice, and a value for $\Gamma(s)$ must be computed as well. The primary contribution to the calculations is the evaluation of n^{-s} in the summations. If these are pre-computed, the Borwein algorithm still wins, as shown in the next figure. All calculations were performed on a contemporary desktop computer, using the GMP library; the algorithms were implemented in the C programming language. Note that the actual performance depends on the underlying implementation of log, exp, sin, atan, sqrt, gamma and the like; the actual implementation used here is fairly naive and untuned

of being complex-valued. Thus, this section will present an experimental study of the performance, based on a reasonably well-developed implementation of the various quantities, anchored upon the basic functions provided by GMP.

These experimental results are shown in Figs. 1 and 2 for the polylogarithm, and Figs. 3 and 4, for the Hurwitz zeta function, which compare the performance of actual implementations.

The first figure shows "cold cache" performance, measured in seconds, as compared to the number of desired decimal places of precision. The figure is termed "cold cache", in recognition of the fact that some constants may be pre-computed. For example, if s is held fixed, while z is varied, then the values of n^{-s} appearing in both the direct sum and the Borwein algorithm may be computed once, and then re-used for subsequent calculations. As the figures indicate, computing n^{-s} can be very expensive for general complex-valued s, and so the caching strategy offers a big gain when s is held constant.

As the "cold cache" figure demonstrates, the Borwein algorithm is faster than direct summation. The problem with direct summation is that it requires more values of n^{-s} to be computed to achieve comparable precision.

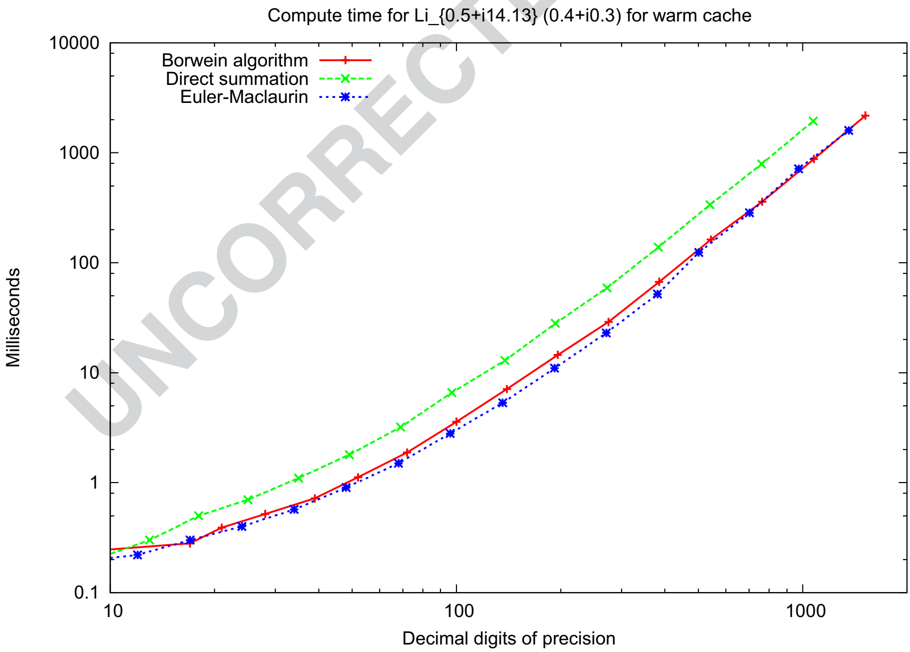

Fig. 2 Polylogarithm warm cache. This graph compares the compute times for the Borwein algorithm, the direct summation of the polylogarithm, and the Euler–Maclaurin series, for a "warm cache". That is, an array of values of n^{-s} were pre-computed, prior to beginning the timing measurements. These pre-computed values were used in the Borwein and direct sums. For the Euler–Maclaurin sum, the value of $\Gamma(s)/(2\pi i)^s$, needed for Jonquière's identity, is pre-computed and cached as well. Despite the use of such cached, pre-computed values, the Borwein algorithm still wins over direct summation, and appears to be tied with the Euler–Maclaurin sum. This graph uses the same s and z values as in the previous graph

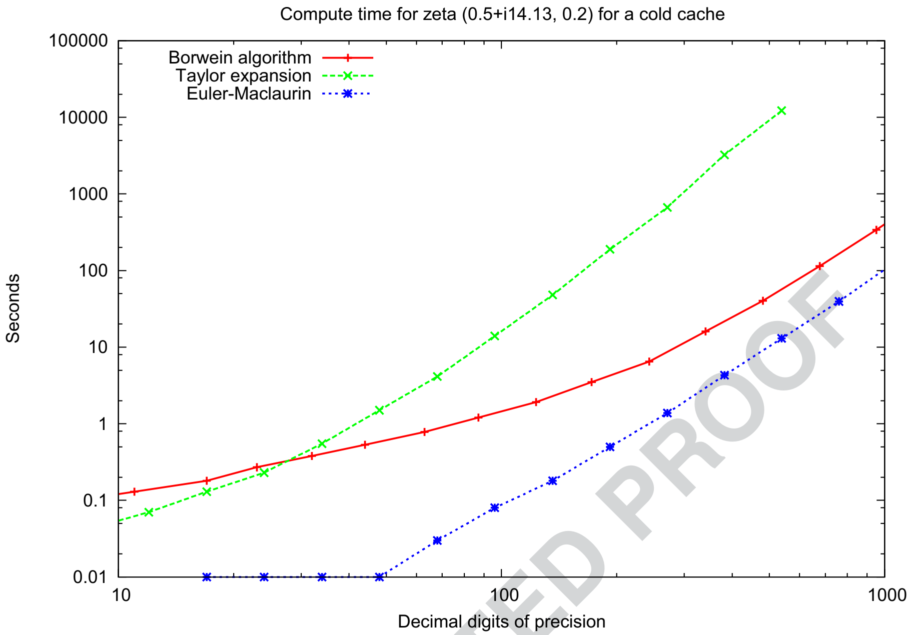

Fig. 3 Hurwitz zeta cold cache performance. This figure compares the performance of the Borwein algorithm, the evaluation of the Taylor expansion, and Euler–Maclaurin summation, for the Hurwitz zeta. The vales computed are for $s = 0.5 + i14.13$ and $q = 0.2$. For such a small value of q, one might have hoped that the Taylor expansion might converge quickly. This appears to not be the case, as the binomial coefficients can grow to be quite large, and thus force many terms to be summed. The figure is for a "cold cache", assuming that no constants have been pre-computed. The lines are not parallel, with the Taylor expansion getting progressively worse. Both the Borwein and the Euler–Maclaurin algorithms are orders of magnitude faster than the Taylor's series for high precisions; the Euler–Maclaurin algorithm appears to be an easy winner for all precisions

As the "warm cache" figure shows, the Borwein algorithm will still win against direct summation, even when the values of the n^{-s} are pre-computed. Even when these are pre-computed, and can be pulled from the cache, direct summation still requires more operations.

A different but equally dramatic set of results obtain, when one compares the performance of the Borwein algorithm applied to the Hurwitz zeta function, as compared to the use of the Taylor's expansion (4) for the Hurwitz zeta. In this case, as the figures show, the Taylor's series outperforms the Borwein algorithm by a constant factor of three, when both algorithms use pre-computed constants. However, the cost of pre-computing these constants skyrockets for the Taylor's series, making it unattractive as the number of required decimal places increases.

Although the Borwein algorithm seems to show a slight advantage over the Euler–Maclaurin series when it is used to evaluate the polylogarithm, much of that advantage disappears when evaluating near the branch point $z = 1$. The Borwein algorithm breaks down near the branch point, and requires the (possibly recursive) application of the duplication formula (given in the next

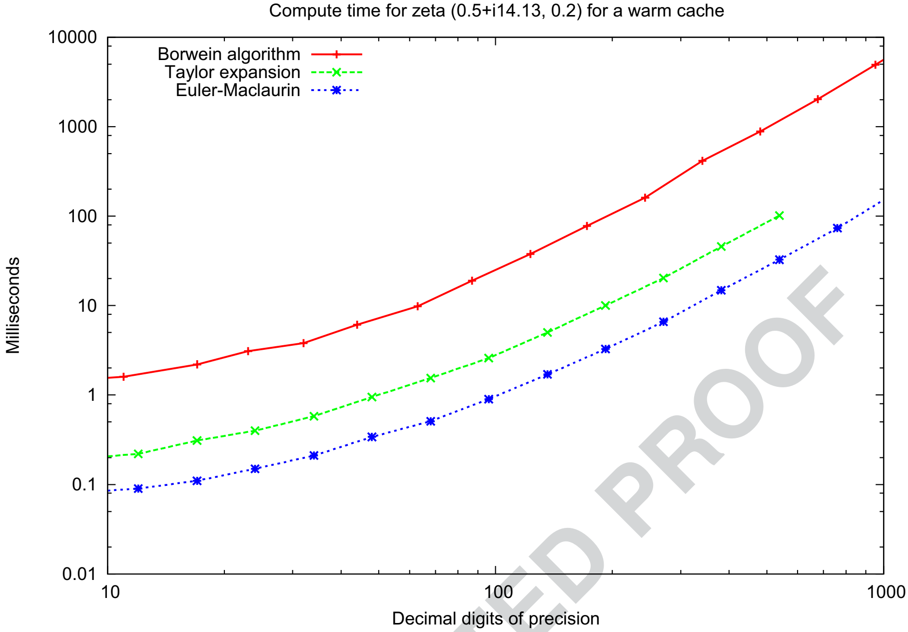

Fig. 4 Hurwitz zeta warm cache performance. This figure compares the evaluation times for Euler–Maclaurin summation, the Borwein algorithm, and the Hurwitz zeta Taylor's expansion, for the "warm cache" scenario. In this case, it is recognized that if s is assumed to be held fixed, then the values of n^{-s} appearing in the Borwein algorithm may be pre-computed. Similarly, the values of $\zeta(s+n)$ and the binomial coefficients $\binom{s+n-1}{n}$ appearing in the Taylor's series and the Euler–Maclaurin formula may be pre-computed. For this value of q, the Borwein algorithm requires four evaluations of the polylogarithm; values of q closer to 0.5 would require only two. The Taylor's series evaluation appears to be about eight times faster than these four evaluations (or four times faster than the minimum of two evaluations). Note that the time axis here is in milliseconds, not seconds. Comparing to the previous graph, it is clear that performing the pre-computations can be terribly expensive. The first evaluation of the function can take 100 or 1,000 times longer than subsequent evaluations at the same value of s

section) to obtains points sufficiently far away from $z = 1$. By contrast, the Euler–Maclaurin series seems to happily converge in this area, and so no extra steps are required.

In conclusion, it appears that the Euler–Maclaurin formula provides the best algorithm for evaluating the Hurwitz zeta, and can often tie and sometimes outperform the Borwein algorithm for the polylogarithm.

9 Duplication formula

The region of applicability of the algorithm may be extended by making use of the duplication formula for the polylogarithm or periodic zeta. The duplication formula, or more generally, the multiplication theorem, is an extension of the well-known Legendre duplication formula for the gamma function [1, (6.1.18), (6.3.8), (23.1.10)], into the domain of the polylogarithm.

Thus, for example, the formula (23) together with the error bounds (24), (25) allow $F(q; s) = \mathrm{Li}_s\left(e^{2\pi iq}\right)$ to be computed for real q in region $1/4 \le q \le 3/4$. To obtain values on the region $0 < q < 1/4$, one applies the duplication formula

$$F(q; s) = 2^{1-s} F(2q; s) - F\left(q + \frac{1}{2}; s\right)$$

recursively until one obtains values of $q \ge 1/4$. Rearranging terms, one obtains a similar formula for iterating values of $q > 3/4$ until they are less than or equal to $3/4$. Thus, as q approaches 0 or 1, the algorithm requires more time, but only logarithmically so, as $-\log_2 q$ or $-\log_2(1-q)$ as the case may be.

The equivalent formulas [20, Sections 7.3.1, 7.12.1] for the polylogarithm are

$$\mathrm{Li}_s(z) + \mathrm{Li}_s(-z) = 2^{1-s}\mathrm{Li}_s\left(z^2\right)$$

whereas the p-adic version has the appearance of a Gauss sum:

$$\sum_{m=0}^{p-1} \mathrm{Li}_s\left(ze^{2\pi im/p}\right) = p^{1-s}\mathrm{Li}_s\left(z^p\right)$$

This last identity is most easily understood to be one form of the so-called *multiplication theorem*, having analogues in the polygamma function, Hurwitz zeta function and the Bernoulli polynomials, all being derivable from Gauss' multiplication theorem for the gamma function, and having applications in both number theory and dynamical systems (Fig. 5, 6, 7, 8, and 9).

The application of the duplication formula, together with the inversion relation (6) can be used to extend the evaluation of the polylogarithm to the entire complex plane. For numerical work, both formulas must be applied, one

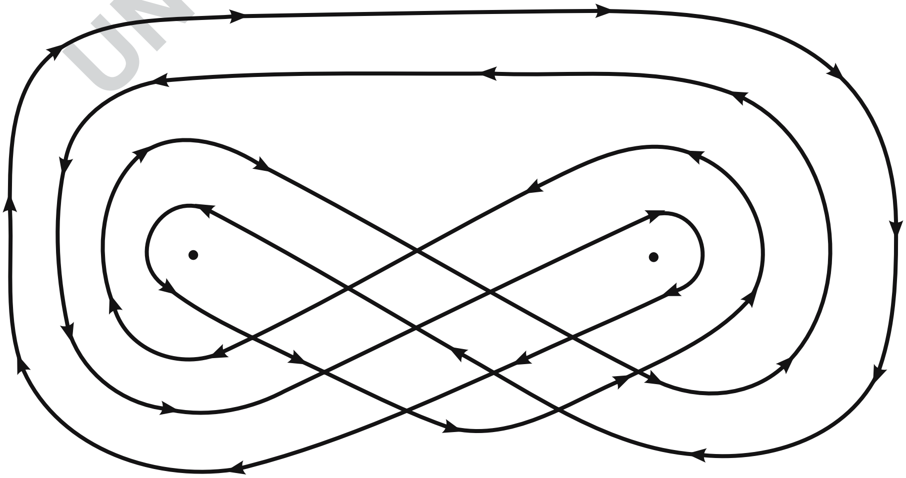

Fig. 5 Polylogarithm monodromy. A graph of the polylogarithm monodromy, given by (32)

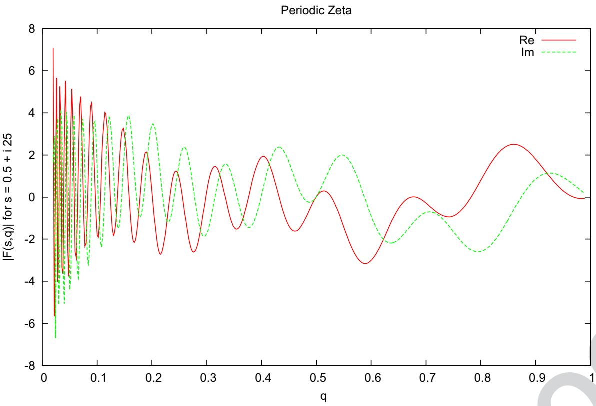

Fig. 6 Periodic zeta function. This graph shows the real and imaginary parts of the periodic zeta function $F(q,s) = \mathrm{Li}_s\left(e^{2\pi iq}\right)$ for $s = \frac{1}{2} + i25$. This value of s is close to a zero of the Riemann zeta function, at $s = \frac{1}{2} + i25.01085758014\ldots$ Thus, both the real and imaginary parts approach zero at $q = 1$, as well as at $q = \frac{1}{2}$. The increasing oscillations as $q \to 0$ are due to the contribution of the very first term of the Hurwitz zeta: that is, these oscillations are nothing other than that of $q^{-s} = e^{i25\log q}/\sqrt{q}$. Subtracting these away is the same as analytically continuing to the region $q > 1$, and matches the coarser set of oscillations, which are given by $(1+q)^{-s} = e^{i25\log(1+q)}/\sqrt{1+q}$ Noteworthy is that the presumed "periodicity" in q is very misleading: the image suggests an essential singularity at $q = 0$, and continuing, logarithmically slower oscillatory behavior for the analytic continuation to the region where $q > 1$

alone is not enough. Consider first the application of the duplication formula only. It may be used to take points that are near to $z = 1$, and map them further away from $z = 1$, into the convergent region for the Borwein polynomial. Repeated application allows arbitrarily close approach to $z = 1$ from the left-hand side. The resulting region of convergence is kidney-shaped, with the cusp of the kidney at $z = 1$, and the kidney containing the unit disk $|z| \leq 1$. The precise shape of the kidney depends on the number of terms one wishes to use in the polynomial approximation. The shape that can be achieved while still maintaining good running time is shown in Fig. 10. However, as can be seen, this strategy barely penetrates the $\Re z > 1$ region.

To extend the algorithm to the entire complex z-plane, one must make use of the Jonquière's inversion formula [23, pp. 27–32]

$$e^{-i\pi s/2}\mathrm{Li}_s(z) + e^{i\pi s/2}\mathrm{Li}_s\left(\frac{1}{z}\right) = \frac{(2\pi)^s}{\Gamma(s)}\zeta\left(1-s, \frac{\log z}{2\pi i}\right) \tag{27}$$

together with some sort of independent means of evaluating the Hurwitz zeta function. The Taylor expansion (4) is particularly well-suited, as it is rapidly convergent in the vicinity of $z = 1$. Specifically, it is convergent when $|\log z| < 2\pi$, although the region of acceptable numerical convergence is smaller, roughly $|\log z| < \pi$. Either way, this encompasses a rather large region in the vicinity of $z = 1$, which is exactly the region where the Borwein algorithm has trouble. The inversion formula is used to move points $|z| > 1$ from outside of the unit circle, to the inside. Not all points inside the unit circle are directly

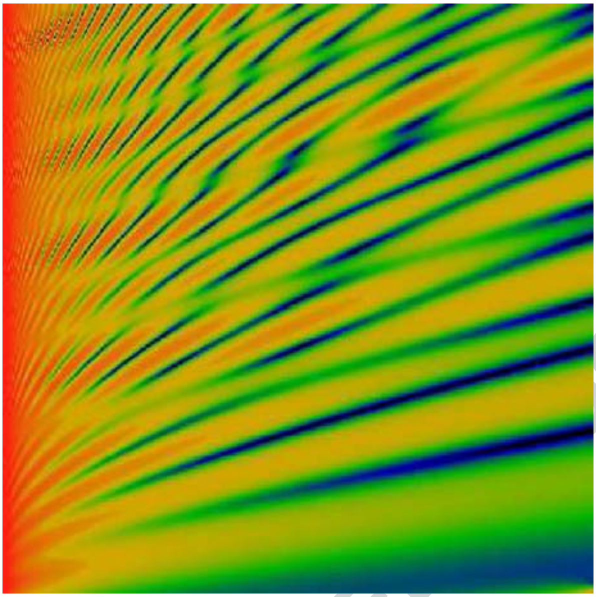

Fig. 7 The periodic zeta. This image illustrates the so-called "periodic zeta function" $F(q;s) = \mathrm{Li}_s\left(e^{2\pi i q}\right) = \sum_{n=1}^{\infty} \frac{\exp(2i\pi nq)}{n^s}$ Graphed is the magnitude of $F(q;s)$, with a color scale such that zero is shown as black, one is a shade of greenish-blue, and two is yellow, and larger values as orange-red. Along the horizontal axis runs q, taken to be real, from $q=0$ on the left to $q=1$ on the right. Then, writing $s=\frac{1}{2}+i\tau$, the value of τ is varied along the vertical axis, running from $\tau=0$ at the bottom of the image, to $\tau=50$ at the top of the image. The non-trivial zeroes of the Riemann zeta function $\zeta(s)=F(1,s)$ are located where the blue lines intersect the right edge of the image. From the bottom, the first three zeroes are at $s=0.5+i14.13\cdots$, $0.5+i21.02\cdots$, $0.5+i25.01\cdots$. Due to the relation to the Dirichlet eta function, the zeros also materialize at the same values of s, but on the $q=1/2$ line. Remarkably, the blue streaks seem to be roughly parabolic, but are interrupted by nearly straight "erasures". The pattern is reminiscent of graphs of the strong-field Stark effect (need ref). In the Stark effect, eigenvalues are given by the characteristic values of the Mathieu functions. These cross over one another in a curious fashion; see for example, Fig. 20.1 in Abramowitz and Stegun [1]. The precise form of the parabolic blue streaks are due to a term of the form q^{-s}, as given by (35). These may be subtracted, as illustrated in the next image

accessible to the Borwein algorithm; the duplication formula must still be used to handle points in the disk interior that are near $z=1$. Similarly, points with very large z, such as those for which $|z|>e^{2\pi}$, require the use of the duplication formula to be moved into the region of convergence for the Hurwitz Taylor series. A typical view of the polylogarithm on the complex z-plane is shown in Fig. 11.

As a practical computational matter, rather than using (27), it seems to be more convenient to use the formula

$$\mathrm{Li}_{1-s}(z) = \frac{\Gamma(s)}{(2\pi)^s}\left[e^{i\pi s/2}\zeta\left(s,\frac{\log z}{2\pi i}\right) + e^{-i\pi s/2}\zeta\left(s,1-\frac{\log z}{2\pi i}\right)\right] \qquad (28)$$

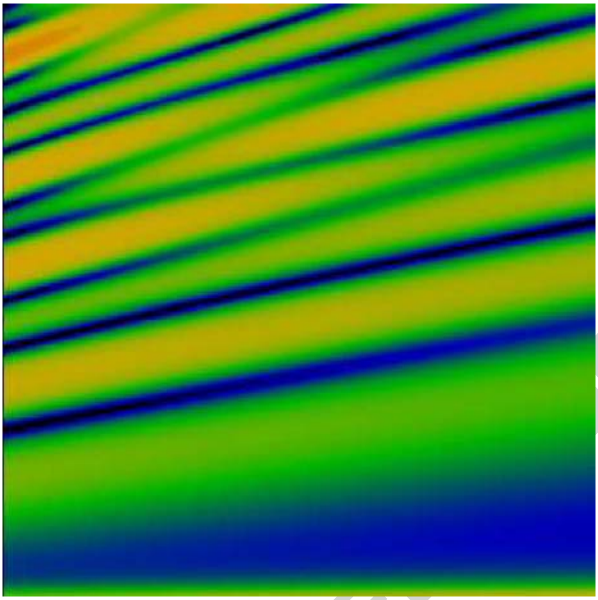

Fig. 8 Hurwitz zeta, with leading term subtracted. This image shows the magnitude of $\zeta(s, q+1) = \zeta(s, q) - q^{-s}$ for the range of $0 \leq q \leq 1$along the horizontal axis, and $0 \leq \tau \leq 50$ along the vertical axis, just as in the previous image; the same coloration scheme is used as well. This essentially corresponds to the previous picture, with the leading divergence subtracted; for large positive τ, the magnitude of the periodic zeta and the Hurwitz zeta differ by an exponentially small amount. A careful comparison of this picture to the previous now shows that the "erasures" or "streaks" in the previous image correspond exactly to the dark parts of this image. Again, a cris-cross lattice pattern can be seen. The dominant streaks in this picture are presumably due to the $(1+q)^{-s}$ term

Although this is equivalent to (27), it correctly captures the appropriate branch of the logarithm that should be used: (28) works well for values of s in both the upper and lower half-planes, whereas (27) is more difficult to correctly evaluate in the lower half s-plane. In either case, one must use the logarithm with an unusual branch cut, taking it to run from $z = 0$ to the right along the positive real axis, as opposed to the usual cut taken for the logarithm.

It is of some curiosity to note that, when $q$is not an integer, the sheet that is forced takes the form

$$\zeta(s, q) + e^{-i\pi s}\zeta(s, 1-q) = \sum_{n=-\infty}^{+\infty} \frac{1}{(n+q)^s}$$

which is also noted by Costin and Garoufalidis [9, Eqn. 14] as a Mittag–Leffler type decomposition [5]. A general discussion of the monodromy is given in a later section.

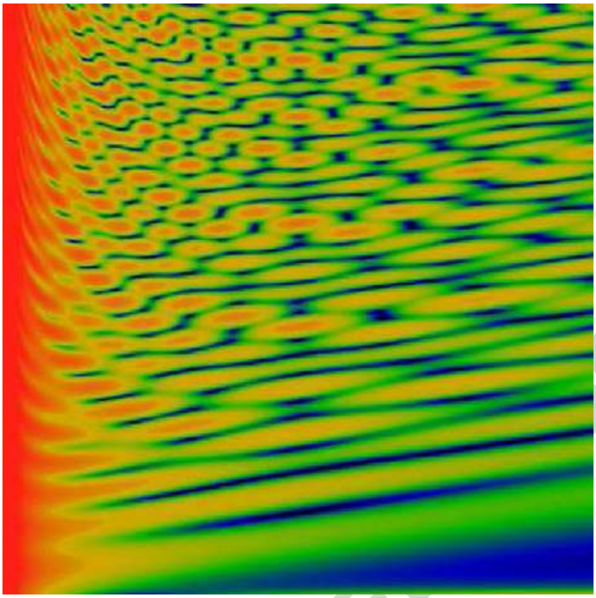

Fig. 9 Extended view of Hurwitz zeta. This figure shows a view of the magnitude $|\zeta(s,q)|$ of the Hurwitz zeta over an extended range, together with a rescaling of coordinates in an attempt to straighten out the striations. The value of q ranges from 0 to $2\sqrt{2} \approx 2.8$ along the horizontal axis, although it is scaled as $q^{3/2}$ along this axis. This rescaling essentially turns the parabolic striations into straight, radial lines emanating from the origin at the lower left. This image also re-scales the τ coordinate, in an attempt to turn the radial striations into horizontals. The value of $q = 1$ is a vertical line running exactly down the middle of the image; the non-trivial Riemann zeroes are visibly stacked up on this line. The value of τ runs from 0 to 100 on this vertical line. On other verticals, τ has been rescaled by $q^{1/2}$, so that τ runs from 0 to 141 on the right hand side of the image, while being essentially held at zero along the left edge of the image. The increasing complexity of the chain-link pattern is apparent as one moves up the imaginary axis. Equally apparent is that the maximum range of excursion of the magnitude changes only slowly over the range of the image: as the pattern increases in complexity, it does not wash out, but has a distinct binary on/off character

10 Testing and validation

The correctness of a given numerical implementation can be validated in a number of ways. For non-positive integer s, one has the exact rational functions previously mentioned. For positive integer n, one has the relationship [20, 30]

$$\mathrm{Li}_n\left(e^{2\pi iq}\right) + (-1)^n \mathrm{Li}_n\left(e^{-2\pi iq}\right) = -\frac{(2\pi i)^n}{n!} B_n(q)$$

where the $B_n(x)$ are the Bernoulli polynomials. For $z = -1$, one regains the Riemann zeta function $\zeta(s)$:

$$\mathrm{Li}_s(-1) = \frac{1}{2^{1-s} - 1}\zeta(s)$$

For $|z| < 1$, the defining series (5) is explicitly convergent, and may be directly summed, thus offering a fourth check of the correctness of an implementation.

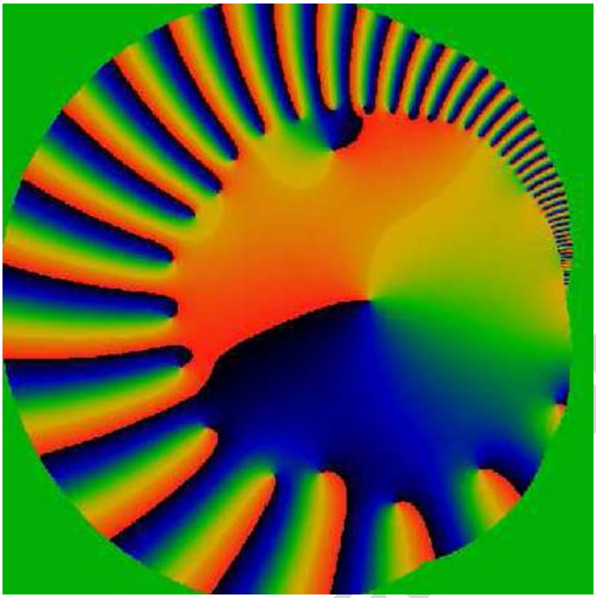

Fig. 10 Polylogarithm phase plot. This image shows a phase plot for the polylogarithm, on the complex z-plane, for $s = 0.5 + i80$. The color scheme is such that black represents the phase $\arg \mathrm{Li}_s(z) = -\pi$, red represents the value of $+\pi$, and green a phase of zero. Points where the phase wraps around the full color range are zeros. Thus, each of the "fingers" in this image terminates on a zero. Most of these zeros are close to, but usually not quite on the circle $|z| = 1$. The lone zero near the center of the image corresponds to the zero at $z = 0$. The slight cusp at the right side of the image corresponds to $z = 1$. The outer boundary of the image, adjoining to the solid green area, represents the limit of convergence for the combination of the Borwein algorithm and the duplication formula; one can enlarge the region of convergence slightly, but not much, but using polynomial approximations of higher order. The boundary is easily crossed by applying the inversion formula, as the later images show. The image is offset somewhat, rather than being centered on $z = 0$, because most of the convergent region is to the left of the imaginary axis. Note the other lone zero floating along inside of the $|z| < 1$ disk. If this picture were animated as τ increased from 0 in the positive direction, then the zero fingers can be seen as marching across in a counter-clockwise fashion, starting at $z = 1$ and proceeding around. Most zeros pass to another sheet upon crossing $z = 1$, after making a full revolution; others spiral around a second time on this sheet, such as the lone zero up top. An animated movie of this image, showing the spiraling, can be seen at http://www.linas.org/art-gallery/polylog/polylog.html

Finally, the multiplication theorem can be used to check for consistency. Each of these quantities can be computed by independent algorithms, and thus be used to validate the correctness of a polylogarithm implementation.

11 Branch points and monodromy

The principal sheet of the polylogarithm has a branch point at $z = 1$, and by convention, a branch cut is placed along the positive real z-axis, extending to the right. As one moves off of the principal sheet, one discovers that there is

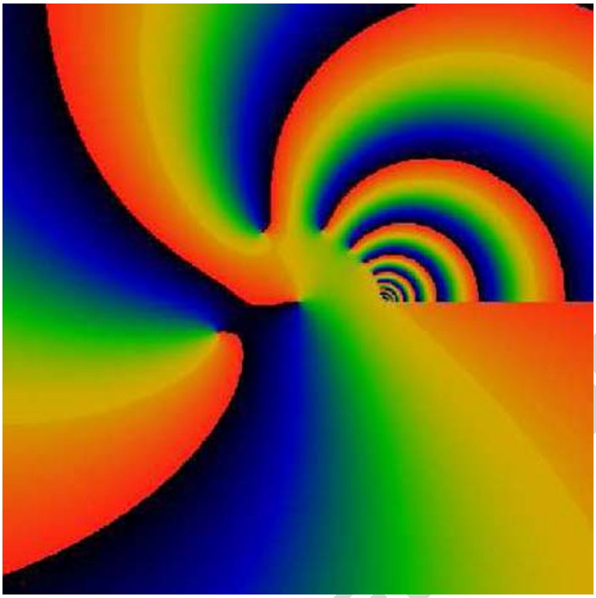

Fig. 11 Polylog whole plane. This image illustrates the results of combining the Borwein algorithm together with the inversion and duplication formulas. The image shows the complex z-plane, is centered on $z = 0$, and runs from -3.5 to $+3.5$, left to right, top to bottom. The value of s is fixed at $s = 0.5 + 15i$ for the whole picture. The color scheme is the same as for the previous picture. Besides the zero at $z = 0$ at the center of the picture, another zero is visible to the left, and slightly down. This zero is an "ex-Riemann zero", in that, if instead one had created the image for $s = 0.5 + i14.1347\cdots$, then this point would have been located exactly at $z = -1$. As it happens, this point has now rotated to slightly below $z = -1$. The zeroes above and to the right of the origin are Riemann zeroes to be: as τ increases, each will in turn rotate counterclockwise to assume the position at $z = -1$. The concentric shells are centered on the point $z = 1$. The branch cut can be seen extending to the right from $z = 1$, on the positive real axis.

another branch point at $z = 0$. The resulting monodromy group is generated by two elements, acting on the covering space of the bouquet $S^1 \vee S^1$ of homotopy classes of loops in $\mathbb{C} \setminus \{0, 1\}$ passing around the branch points $z = 0$ or $z = 1$. The author is not aware of any simple published discussion of the monodromy for the polylogarithm; a much more abstract discussion is given in [4, 9, 15, 21]. Thus, this section provides a discussion; note that the correct manipulations to move from one sheet to another can be somewhat treacherous and confusing.

The inversion formula (27) suggests a way to move around the branch point at $z = 1$. Suppose one starts at the point $z = x + i\varepsilon$ with x real, positive, and greater than one, and ε some arbitrarily small positive real number; thus z is very near the branch cut of the principal sheet. One wishes to compare this to the neighboring value on the other side of the cut, at $x - i\varepsilon$. Applying the inversion formula, one can bounce these two points inside the unit circle, where there is no cut, and thus the polylog differs by $\mathcal{O}(\varepsilon)$. The difference across the cut is thus

$$\begin{aligned}\Delta &= \mathrm{Li}_s(x+i\varepsilon) - \mathrm{Li}_s(x-i\varepsilon) \\ &= e^{i\pi s/2}\frac{(2\pi)^s}{\Gamma(s)}\left[\zeta\left(1-s, \frac{\ln x + i\varepsilon}{2\pi i}\right) - \zeta\left(1-s, \frac{\ln x - i\varepsilon}{2\pi i}\right)\right] + \mathcal{O}(\varepsilon)\end{aligned}$$

Now, since $x > 1$, a naive application of this formula yields $\Delta = 0$ since $\log(x + i\varepsilon) - \log(x - i\varepsilon) = \mathcal{O}(\varepsilon)$; but clearly Δ cannot be zero. To resolve this situation, one must make the Ansatz that the cut of the logarithm should be crossed. This may be done by taking the cut of the logarithm to extend to the right, instead of to the left, as it is by convention. One then gets

$$\log(x+i\varepsilon) - \log(x-i\varepsilon) = 2\pi i N + \mathcal{O}(\varepsilon) \tag{29}$$

for some integer N. The difference across the cut, for positive N, is then

$$\zeta(s,q) - \zeta(s, N+q) = \sum_{k=0}^{N-1}\frac{1}{(k+q)^s}$$

with $q = \ln(x)/2\pi i$. By swinging the cut of the logarithm so that it extends to the right, one obtains that the real part of q is positive; the real part of q runs between 0 and 1. As a result, the value of $(k+q)^s$ is unambiguous; as otherwise taking something to the power s also begs the question of which sheet the power must be taken on. That is, for general complex w, the power function is multi-sheeted:

$$w^s = e^{s\ln w} \to e^{s(\ln w + 2\pi i M)} = e^{2\pi i s M} w^s \tag{30}$$

for some integer M representing the sheet of the power function. Since the real part of q is taken as positive, one can temporarily operate with the assumption that $M = 0$ (Fig. 12). Q2

Taking $N = 1$, the above reasoning provides an excellent description, which may be verified numerically. The concentric semi-circles visible in the image (11) can be entirely explained by the behavior of q^s as $q \to 0$, that is, as $z \to 1$. Thus, the difference between the $N = 1$ sheet and the $N = 0$ sheet is

$$\Delta_1 = e^{i\pi s/2}\frac{(2\pi)^s}{\Gamma(s)}\left(\frac{\ln z}{2\pi i}\right)^{s-1} \tag{31}$$

Properly speaking, Δ_1 is a function of s and z, and so one should write $\Delta_1(s; z)$ to signify this. However, for ease of notation, this marking is dropped, and is taken implicitly in what follows. The difference $\mathrm{Li}_s(z) - \Delta_1$ is illustrated in Fig. 13; the concentric rings are seen to be canceled out precisely, leaving behind a smoothly varying function, having the expected smooth structure. Moving across the cut, for $\Re z > 1$, the joint appears to be smooth. For general positive N, it is easy to confirm numerically that the discontinuity between the N'th sheet, and the $N-1$'th sheet, across the $\Re z > 1$ cut is

$$\Delta_N = e^{i\pi s/2}\frac{(2\pi)^s}{\Gamma(s)}\left(N - 1 + \frac{\ln z}{2\pi i}\right)^{s-1}$$

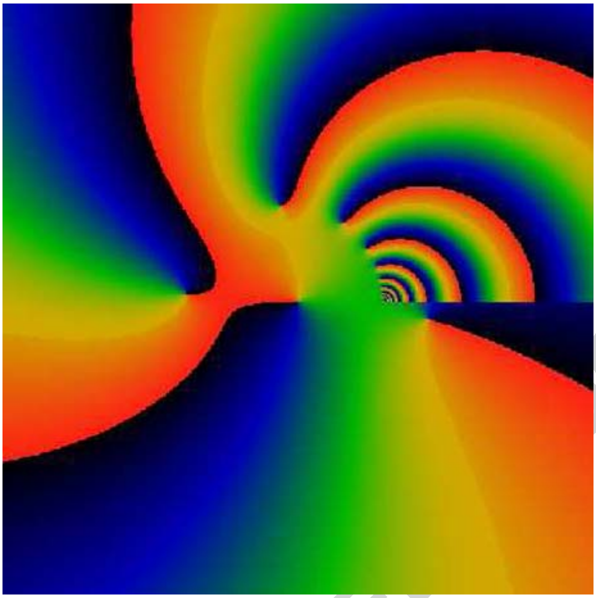

Fig. 12 Polylogarithm off the critical axis. This figure shows the polylogarithm in the complex z-plane, at fixed $s = \sigma + i\tau = 1.2 + i14$. It is superficially similar to the previous image, with two notable differences. Visible just under the cut, on the right, is a zero. If σ were smaller, this zero would move to the left; at $\sigma = 0.5$, it would be very near to the branch point at $z = 1$. As τ gets larger, this point moves up, crossing the branch cut and moving onto the next sheet. Of course, if instead one had $\sigma + i\tau = 0.5 + i14.13\ldots$, this zero would be located precisely at $z = 1$. Thus, this figure illustrates what a Riemann zero looks like when it is not on the critical strip. As τ increases, the remaining zeroes circle around in a widening spiral: thus, the zero at the far left of the image is not near $z = -1$, but is to the left of there (and thus is not going to be a zero of the Dirichlet eta function). If instead one had σ with a value of less than 0.5, the zeroes would spiral around in an ever-tightening spiral, and would never hit the branch cut or cross over. The Riemann hypothesis is essentially that these zeroes are impacting exactly at the branch point; a violation of RH would be a pair of zeroes that failed to hit the branch point, instead passing to the left and right of the branch point. This figure suggests that RH holds: the zeroes are clearly lined up in single file; it is hard to imagine how this single file might break ranks into a pair of zeros that simultaneously passed near the branch point

The discontinuity across the cut $\Re z > 1$, for negative N, follows similarly. Next, taking $N = -1$ in (29), (or $N = 0$ in the equation immediately above, which can serve as a point of confusion) the difference is given by -1/$(-1 + q)^{1-s} = e^{-i\pi s}/(1-q)^{1-s}$, or

$$\Delta_{-1} = e^{-i\pi s/2} \frac{(2\pi)^s}{\Gamma(s)} \left(1 - \frac{\ln z}{2\pi i}\right)^{s-1}$$

Figure 14 illustrates the difference $\mathrm{Li}_s(z) - \Delta_{-1}$. For general negative N, the difference between adjacent sheets is then

$$\Delta_{-N} = e^{-i\pi s/2} \frac{(2\pi)^s}{\Gamma(s)} (N - q)^{s-1}$$

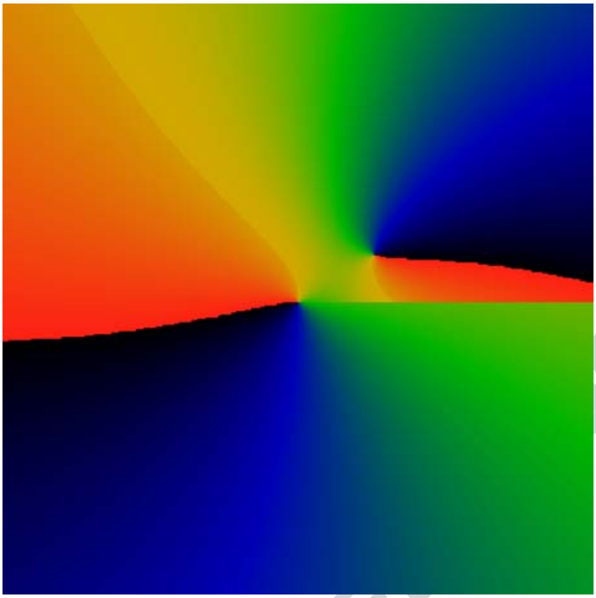

Fig. 13 Upper sheet of the polylogarithm. This figure shows the $g_1 \mathrm{Li}_s(z) = \mathrm{Li}_s(z) - \Delta_1$ sheet of the polylogarithm on the complex z-plane, obtained by circling the branch-point at $z = 1$ in the right-handed direction. Two zeroes are visible: one at $z = 0$, and the other an "ex-Riemann zero", located not far from $z = 1$, a bit above the real axis. For this image, s is held constant, at $s = 0.5 + i15$; the Riemann zeta function has its first non-trivial zero at $s = 0.5 + i14.13\ldots$. At that value, the above zero would have been located precisely at $z = 1$, on the branch point of the polylogarithm. As τ increases, the Riemann zeroes pass precisely through this branch point, leaving the first sheet and moving to this one. Indeed, the Riemann hypothesis can be taken as being equivalent to the statement that the Riemann zeroes always pass through the branch point in moving from one sheet to another. Taking $s = \sigma + i\tau$, if σ is held constant at $\sigma < 1/2$ while τ is increased, the polylogarithm zeroes never pass through the branch cut, and stay on the main sheet; whereas for $\sigma > 1/2$, they do pass through the branch cut, but not at the branch point

Again, it can be numerically verified that the transition from sheet to sheet is smooth as one crosses the cut $\Re z > 1$.

It is perhaps more clear to use explicit topological language. Let m_1 represent the homotopy class of all loops based at some point z on the complex plane, that wind once around the branch-point $z = 1$ in the positive direction. The action of m_1 on the polylogarithm has the effect of carrying the polylog from one sheet to the next. In the above discussion, it was determined that

$$m_1 \cdot \mathrm{Li}_s(z) = \mathrm{Li}_s(z) - \Delta_1$$

The logarithm in Δ_1 has a branch point at $z = 0$. That is, after acting with m_1, one is now on a sheet with a cut extending from $z = 0$ to the right. Let let m_0 represent the homotopy class of loops that wind once around the branch point $z = 0$ in a right-handed fashion. Acting on the logarithm, one has

$$m_0 \cdot \ln z = \ln z + 2\pi i$$

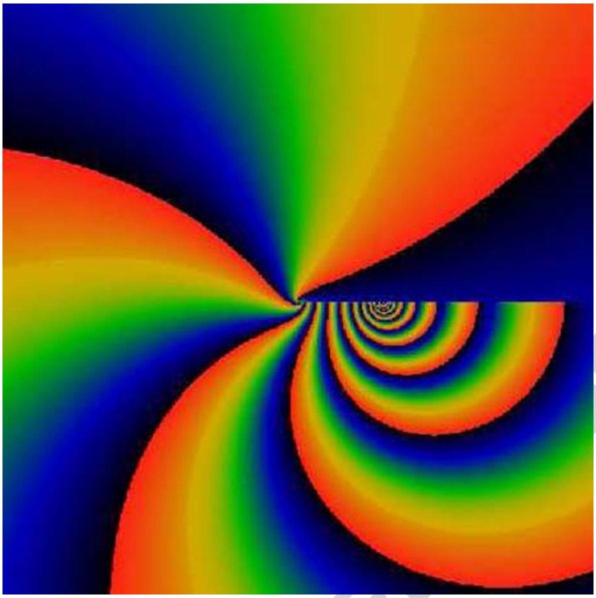

Fig. 14 Lower sheet of the polylogarithm. This figure shows the $g_1^{-1}\mathrm{Li}_s(z) = \mathrm{Li}_s(z) - \Delta_{-1}$ sheet of the polylogarithm on the complex z-plane, obtained by going around the branch-point at $z = 1$ in the left-handed (clockwise) direction. The concentric curves are centered on $z = 1$, and whose form is essentially that of $(1 - q)^{1-s}$. The vertex of the black triangle approaches $z = 0$. By visual inspection, it is clear how to glue this sheet to the principal sheet shown in Fig. 11. After this gluing, a cut remains between the points $z = 0$ and $z = 1$. This cut may be glued to the sheet that results by winding around the branch at $z = 0$ in a clockwise manner. The result is shown in the next figure

Recalling the definition of $q = \ln z / 2\pi i$, one thus has $m_0 \cdot q = q + 1$, and so

$$m_0 \cdot \Delta_N = \Delta_{N+1}$$

The principal sheet of the polylogarithm has no branch point at $z = 0$, and so one has

$$m_0 \cdot \mathrm{Li}_s(z) = \mathrm{Li}_s(z)$$

Winding in the opposite direction, one has

$$m_1^{-1} \cdot \mathrm{Li}_s(z) = \mathrm{Li}_s(z) - \Delta_{-1}$$

In order for m_1 to be properly considered as the group-theoretic inverse of m_1, one must have $m_1 \cdot m_1^{-1} \cdot \mathrm{Li}_s(z) = m_1^{-1} \cdot m_1 \cdot \mathrm{Li}_s(z) = \mathrm{Li}_s(z)$. The resolution of this implies $m_1 \cdot \Delta_{-1} = -\Delta_1$, which in turn implies that $m_1 \cdot q = q + 1$. This seems strange, as the logarithm has no branch point at $z = 1$, and so there is nothing to wind around. By this argument, one would have expected that m_1 had no effect on q at all. The point is subtle and is worth establishing clearly; it is the joining of the polylogarithm to the logarithm cuts that causes the effect. Consider only the logarithm (not the polylogarithm), arranged so that the branch extends to the right. Consider starting just below the real axis, and winding around in a right-handed fashion around the point $z = 1$, and finishing just above the real axis. There is no obstruction at $z = 1$, and so this loop can be shrunk to a very short line segment. None-the-less, the short line segment crosses the cut, and its effect is to move to a different sheet. It is for this reason that one has $m_1 \cdot q = q + 1$. A different set of group generators will be defined below, that do capture the idea that winding around $z = 1$ should have no effect on q.

One can now consider the task of more complex paths from sheet to sheet. Passing twice around the $z = 1$ branch, one has

$$m_1^2 \cdot \mathrm{Li}_s(z) = m_1 \cdot \left[\mathrm{Li}_s(z) - \Delta_1\right] = \mathrm{Li}_s(z) - \Delta_1 - \Delta_2$$

Repeating this gluing n times, so that each time, one pastes the sheets so that crossing the $z > 1$ cut is smooth and differentiable, gives

$$\begin{aligned} m_1^n \cdot \mathrm{Li}_s(z) &= \mathrm{Li}_s(z) - \sum_{k=1}^{n} \Delta_k \\ &= \mathrm{Li}_s(z) - e^{i\pi s/2} \frac{(2\pi)^s}{\Gamma(s)} \sum_{k=1}^{n} \frac{1}{(k-1+q)^{1-s}} \\ &= \mathrm{Li}_s(z) - e^{i\pi s/2} \frac{(2\pi)^s}{\Gamma(s)} \left[\zeta\left(1-s, \frac{\ln z}{2\pi i}\right) - \zeta\left(1-s, n + \frac{\ln z}{2\pi i}\right)\right] \end{aligned}$$

which is recognizable from the initial considerations.

To capture the idea of there being no obstruction at $z = 1$ for the logarithm, one may define a group element $g_1 = m_1 m_0^{-1}$, so that one has $g_1 \cdot \ln z = \ln z$. Then, define $g_0 = m_0$. In terms of these generators, one has the relations

$$\begin{aligned} g_0 \cdot q &= q + 1 \\ g_1 \cdot q &= q \\ g_0 \cdot \mathrm{Li}_s(z) &= \mathrm{Li}_s(z) \\ g_1 \cdot \mathrm{Li}_s(z) &= \mathrm{Li}_s(z) - \Delta_1 \\ g_0 \cdot \Delta_N &= \Delta_{N+1} \qquad \text{for } N > 0 \\ g_1 \cdot \Delta_N &= \Delta_N \end{aligned}$$

To complete the picture, one needs the action of g_0 on Δ_{-1}, or equivalently g_0^{-1} on Δ_1. There is some ambiguity, and thus, room for confusion, as the term $(-1)^s$ arises, which may be taken as $e^{i\pi s}$ or as $e^{-i\pi s}$, with the last two inequivalent for non-integer s. The resolution lies in considering the joining of sheets across the cut that runs between $0 < \Re z < 1$. Visually, the cut can be clearly seen in Fig. 14. Consider now the task of gluing sheets across this cut. For a point z just above the line connecting $z = 0$ and $z = 1$, one has $q = \varepsilon + iv$ for some small, positive ε and some general, positive v. Just below this line, one has $q = 1 - \varepsilon + iv$. That is, q differs by 1, of course. This cut cannot be glued to the polylog, of course; it must be glued to another sheet of the logarithm. The correct gluing, for $z = x$ real, $0 < x < 1$, is given by

$$\lim_{\varepsilon \to 0} \left[\Delta_1 (x + i\varepsilon) + e^{i2\pi s} \Delta_{-1} (x - i\varepsilon) \right] = 0$$

That is,

$$g_0 \cdot \Delta_{-1} = -e^{-i2\pi s} \Delta_1$$

This somewhat strange form is nothing more than an accounting trick; it is the result of providing a definition of Δ_{-N} that had a "natural" normalization, avoiding a minus sign. The price of that definition is this glitch. In all other respects, the homotopy proceeds as expected, so that, for example,

$$g_0 \cdot \Delta_{-N} = \Delta_{-N+1}$$

for $N > 1$.

The free combinations of powers of the two operators g_0 and g_1 (or m_0 and m_1) generate a group, the monodromy group of the polylogarithm. For $s = m$ a positive integer, the monodromy group has a finite-dimensional representation of dimension $m + 1$. Well-known is the case for $s = 2$, the dilogarithm, where the representation forms the discrete Heisenberg group. This may be seen as follows. For $s = 2$, one has

$$\Delta_n = 2\pi i \left(\ln z + (n - 1) 2\pi i \right)$$

Repeated applications of g_0 and g_1 result in a linear combinations of $\mathrm{Li}_2(z)$, $\ln z$ and 1; no terms of any other type appear. One may take these three elements as the basis of a three-dimensional vector space. Writing the basis as column vectors, the representation is

$$4\pi^2 \mapsto e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$-2\pi i \ln z \mapsto e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\mathrm{Li}_2(z) \mapsto e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

so that the action may be represented as

$$g_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad g_0 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

These two matrices may be seen to be the generators of the discrete Heisenberg group $\mathscr{H}_3(\mathbb{Z})$. The general group element of the discrete Heisenberg group is

$$\begin{bmatrix} 1 & a & b \\ 0 & 1 & c \\ 0 & 0 & 1 \end{bmatrix}$$

for integers a, b, c. By convention, one writes x for g_0 and y for g_1, and defines a new element z (having no relation at all to the argument z of the dilogarithm) as the commutator $z = xyx^{-1}y^{-1}$, so that

$$z = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The element z is in the center, so that it commutes with x and y, so $zx = xz$ and $zy = yz$. These relations may be taken as giving the group presentation, so that

$$\mathscr{H}_3(\mathbb{Z}) = \left\langle (x, y) \mid z = xyx^{-1}y^{-1},\ zx = xz,\ zy = yz \right\rangle$$

Curiously, one should note that the discrete Heisenberg group has an alternate presentation that is reminiscent of the braid group. This is perhaps not a complete surprise, as a monodromy can be generally understood to be a subgroup of the braid group or mapping class group of the punctured disk. The mapping class group permutes the punctures of the disk in a continuous fashion, so that the result is an intertwining of strands. For a disk with two punctures, the braid group is B_3 consists of three strands. In this case, two of the strands can be visualized as two parallel lines, each representing one of the two branch points of the polylogarithm. The third strand is the monodromy path, which weaves about the two straight lines, but must always return to the middle. Now, the braid group B_3 has two group generators, denoted σ_1 and σ_2 by convention, which denote a single twist of the left or right pair of strands of the braid. To weave the middle strands of the monodromy about the two branch points, one identifies $g_0 \to \sigma_1^2$ and $g_1 \to \sigma_2^2$. At this point, ignoring the polylog, and limiting ones attention to the universal covering space for the homotopy group over two branch points, one very simply has that g_0 and g_1 are the generators of the free group in two letters. Now for the curious resemblance. The braid group B_3 has the group presentation $\sigma_1\sigma_2\sigma_1 = \sigma_2\sigma_1\sigma_2$. The discrete Heisenberg group has the remarkably similar identity $xyyx = yxxy$.

For $s = 3$, one may write

$$g_0 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad g_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

for the basis

$$4\pi^3 i \mapsto e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$2\pi^2 \ln z \mapsto e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$-\pi i \, (\ln z)^2 \mapsto e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\mathrm{Li}_3 \, (z) \mapsto e_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

The generated group is not the four-dimensional Heisenberg group, as there simply aren't enough generators for that. Its not a representation of the three-dimensional Heisenberg group, as the commutator $w = g_0 g_1 g_0^{-1} g_1^{-1}$ is not in the center, since $w g_0 \neq g_0 w$. However, one does have $w g_1 = g_1 w$, and from this one may deduce that the monodromy group has the presentation

$$M = \left\langle g_0, g_1 \,|\, g_1 g_0 g_1^{-1} g_0^{-1} = g_0 g_1^{-1} g_0^{-1} g_1 \right\rangle \tag{32}$$

In any case, the group is solvable and thus thin.

The Heisenberg group may be regained as the quotient group $\mathscr{H}_3(\mathbb{Z}) = M/\langle w \rangle$ where $\langle w \rangle$ is the conjugate closure of w:

$$\langle w \rangle = \left\{ g w g^{-1} \,|\, g \in M \right\}$$

The conjugate closure is, of course, a normal subgroup of M. Since g_1 already commuted with w, what the quotient group construction does is to impose $\langle w \rangle g_0 = g_0 \langle w \rangle$, which shows up on the cosets as the other, "missing" relationship needed to get the Heisenberg group. It is thus that the presentation of the Heisenberg group is regained. The general form of an element $h \in \langle w \rangle$ may be taken to be

$$h = g_0^{-n} w g_0^n = \begin{bmatrix} 1 & 0 & 0 & 2n+1 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

for integer n. Thus, it is clear that $\langle w \rangle$ is abelian, and that $\langle w \rangle \cong \mathbb{Z}$.

For general integer $s = m$, the monodromy group is always unipotent, and thus a nilpotent group. The generators take the form

$$g_0 = \begin{bmatrix} & & & 0 \\ & C & & \vdots \\ & & & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix} \quad \text{and} \quad g_1 = \begin{bmatrix} 1 & 0 & \cdots & & 0 \\ 0 & 1 & & & \vdots \\ \vdots & & \ddots & 0 & 0 \\ & & 0 & 1 & 1 \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix}$$

where C are the binomial coefficients written in an upper-triangular fashion (thus, for example, for $m = 4$, one adds a column of $(1, 3, 3, 1)$ and, for $m = 5$, another column $(1, 4, 6, 4, 1)$, etc.). This form is determined by taking as the basis vectors the monomials appearing in the expansion of $(\ln z + 2\pi i)^m$ (whence the binomial coefficients), and normalizing with a factor of $2\pi i / \Gamma(m)$. The generated monodromy group is isomorphic to the above $s = 3$ case, as it has the same presentation. The element $w = g_0 g_1 g_0^{-1} g_1^{-1}$ commutes with g_1, just as before; but $g_0 w \neq w g_0$. Thus, the conjugate closure $\langle w \rangle$ is again a one-dimensional abelian group, that is, $\langle w \rangle \cong \mathbb{Z}$, and the resulting quotient group is again the Heisenberg group (Fig. 15). Q2

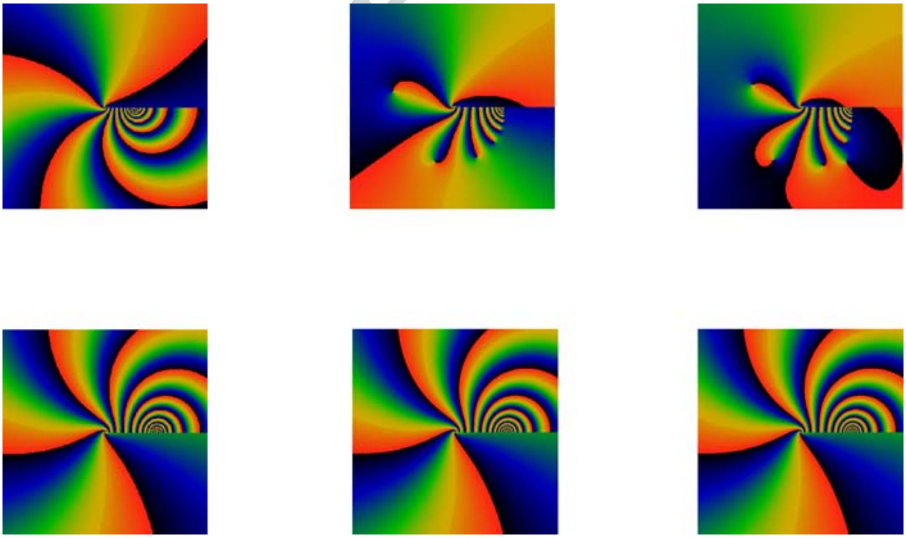

Fig. 15 More polylogarithm sheets. Top row: shows the result of winding around $z = 1$, one, two and three times in a left-handed fashion. The cuts to the right of $z = 1$ join together smoothly from image to image. Using the monodromy group notation, these are the g_1^{-1}, the $g_1^{-1} g_0^{-1} g_1^{-1}$, and the $g_1^{-1} g_0^{-1} g_1^{-1} g_0^{-1} g_1^{-1}$ sheets. Second row: Result of winding left-handed around $z = 1$, followed by winding around $z = 0$ once. The cut $0 < z < 1$ joins smoothly to the images in the row above. The three images appear to be visually indistinguishable; but in fact they are not the same. Using the monodromy group notation, these are the $g_0 g_1^{-1}$, the $g_0 g_1^{-1} g_0^{-1} g_1^{-1}$, and the $g_0 g_1^{-1} g_0^{-1} g_1^{-1} g_0^{-1} g_1^{-1}$ sheets Q3

For s not an integer, the action of g_0 does not close, and the general vector-space representation is infinite dimensional. The action of the g_0 monodromy can be understood to be the shift operator on an infinite-dimensional vector space, whose basis vectors may be taken to be Δ_N; the operator g_1 acts to inject into this shift space. Defining w as before, it is not hard to determine that $w \cdot \Delta_N = \Delta_N$ and that $w \cdot \mathrm{Li}_s(z) = \mathrm{Li}_s(z) + \Delta_1 - \Delta_2$. Remarkably, the same commutation relation holds as before, in that $g_1 w \cdot \mathrm{Li}_s(z) = w g_1 \cdot \mathrm{Li}_s(z) = \mathrm{Li}_s(z) - \Delta_2$. This, together with the more trivial result $g_1 w \cdot \Delta_N = w g_1 \cdot \Delta_N = \Delta_N$ shows that the infinite-dimensional vector space representation of the monodromy has the same presentation as the finite-dimensional case. Thus, one concludes that, for all complex values of $s \neq 2$, the monodromy of the polylogarithm is given by (32).

The analog of the Dirichlet L-functions are functions of the form

$$p^{-s} \sum_{m=1}^{p-1} \chi(m) \, \mathrm{Li}_s\left(z e^{2\pi i m/p}\right)$$

where $\chi(m)$ is a primitive character modulo p. These functions have the strange property of having branch points at $e^{2\pi i m/p}$ whenever $\chi(m)$ is not zero. The resulting monodromy groups have a more complex structure as well.

12 The "periodic" zeta function

The ability to compute the polylogarithm allows one to visualize some of the interesting sums that occur in number theory. One such example is the so-called "periodic zeta function", defined by Apostol [2, Thm. 12.6] as

$$F(q; s) = \sum_{n=1}^{\infty} \frac{e^{i2\pi nq}}{n^s} \tag{33}$$

Clearly, one has $F(q, s) = \mathrm{Li}_s\left(e^{2\pi i q}\right)$. The periodic zeta function occurs when one attempts to state a reflection formula for the Hurwitz zeta function, as

$$\zeta(1 - s, q) = \frac{\Gamma(s)}{(2\pi)^s} \left[e^{-i\pi s/2} F(q; s) + e^{i\pi s/2} F(1 - q; s) \right] \tag{34}$$

What makes the periodic zeta function interesting is that *it is not actually periodic*. That one might think it is seems obvious from the definition (33): the direct substitution of $q \to q + 1$ gives the false suggestion of periodicity in q. This is false because, in fact, $\mathrm{Li}_s(z)$ has a branch point at $z = 1$. The "periodic" zeta function is multi-sheeted, and, attempting to trace a continuous path from $q \to q + 1$, while keeping q real carries one through the branch-point, and from one sheet to the next. This is illustrated graphically, in Fig. 6.

The Fig. 6 shows an oscillatory behavior that clearly diverges as $q \to 0$. This oscillation is can be simply explained. Substituting $s = \frac{1}{2} + i\tau$ into (34) gives

$$\zeta\left(\frac{1}{2} - i\tau,\, q\right) = \frac{e^{-i\pi/4} e^{-i\tau \log 2\pi} e^{i\phi}}{\sqrt{2\cosh \tau\pi}} \left[e^{\pi\tau/2} F(q; s) + i e^{-\pi\tau/2} F(1-q; s)\right]$$

after the substitution of

$$\Gamma\left(\frac{1}{2} + i\tau\right) = \frac{\sqrt{\pi}\, e^{i\phi}}{\sqrt{\cosh \tau\pi}}$$

where ϕ is a purely real phase that can be expressed as a somewhat complicated sum [1, eqn 6.1.27]. For reasonably large, positive τ, such as $\tau = 25$ in the picture, one may ignore the second term, so that the whole expression simplifies to

$$F\left(q; \frac{1}{2} + i\tau\right) = \exp i\left(\frac{\pi}{4} - \phi + \tau \log 2\pi\right)\, \zeta\left(\frac{1}{2} - i\tau,\, q\right) + \mathcal{O}\left(C^{-\tau}\right)$$

for some real constant $C > 0$. Then, as $q \to 0$, the leading contribution to the Hurwitz zeta comes from the $n = 0$ term in (1): that is, $q^{-s} = e^{i\tau \log q}/\sqrt{q}$, so that

$$F\left(q; \frac{1}{2} + i\tau\right) \approx \frac{e^{i\tau \log q} e^{i\psi}}{\sqrt{q}} \quad \text{as } q \to 0 \tag{35}$$

for some fixed, real phase ψ that is independent of q. As is eminently clear from both the picture (6), and from the approximation (35), the limit of $q \to 0$ of $F(q, s)$ cannot be taken: this is the branch-point of $\mathrm{Li}_s(z)$ at $z = 1$.

Curiously, the limit $q \to 1$ does exist, and one has $F(1, s) = \zeta(s)$ the Riemann zeta function. The approximation (35) hints at how to move through the branch point, from one sheet to the next: (35) is the discontinuity between sheets. That is, one has, for large, positive τ

$$F\left(q + 1;\, \frac{1}{2} + i\tau\right) \approx F\left(q,\, \frac{1}{2} + i\tau\right) - \frac{e^{i\tau \log q} e^{i\psi}}{\sqrt{q}}$$

as the formula that approximates the movement from one sheet to the next. In essence, this shows that the "periodic" zeta function is not at all periodic: $F(q + 1; s) \neq F(q; s)$ whenever $\Re s \leq 1$.

The complete relationship between the Hurwitz zeta and the periodic zeta is rather subtle. For example, if instead one considers large *negative* τ, and graphs the periodic zeta, one obtains what is essentially the left-right reversed image of (6), with oscillations approaching a singularity at $q \to 1$. One may easily *numerically* verify that these oscillations are precisely of the form $(1 - q)^{-s}$. From this, one may deduce that the correct form of the reflection formula is

$$F(q, 1 - s) = \frac{\Gamma(s)}{(2\pi)^s} \left[e^{i\pi s/2} \zeta(s, q) + e^{-i\pi s/2} \zeta(s, 1 - q)\right] \tag{36}$$

which captures the oscillatory behavior at $q \to 0$ for large positive τ and the oscillations at $q \to 1$ for large negative τ.

The constraint of keeping q real causes integer values of q to correspond precisely to the branch point of the polylogarithm. This makes reasoning about the continuity of the periodic zeta at integral values of q rather turbid. Since the Riemann zeta function lies precisely at the branch point, this seems to also make it difficult to gain new insight into the Riemann zeta in this way.

13 Conclusion

Both the Borwein and the Euler–Maclaurin algorithms appears to offer a stable and fast way of computing the Hurwitz zeta function for general complex s and real q, and the polylogarithm for general complex s and z. The Euler–Maclaurin algorithm offers superior performance for the Hurwitz zeta. An actual implementation using a variable-precision library shows that all of the algorithms are quite tractable.

An unexplored area is the optimal implementation of the algorithms using standard IEEE double-precision floating-point mathematics. Preliminary work shows that some of the intermediate terms are just large enough so that rounding error may be of concern; the region of high-precision convergence has not been characterized. The propagation of rounding errors through the algorithm has not been characterized; it may be possible to re-arrange terms to minimize the propagation of errors. Such a characterization is necessary before reliable double-precision code can be created. By contrast, such considerations can be swept under the rug when employing arbitrary-precision arithmetic.

For the Borwein algorithm, evaluating the bounds (24) and (25), so as to obtain an order estimate, is non-trivial, and can account for a significant amount of compute time. A speedier heuristic is desirable. By contrast, the Euler–Maclaurin algorithm seems to have a very simple heuristic that is quite adequate for general-purpose applications. In either case, it would be desirable to have a more refined analysis and presentation of the heuristics.

It may be possible to find even more rapidly converging algorithms, by generalizing some of the results from Cohen et al. [26], or by appealing to the general theory of Padé approximants. Both approaches seem promising, although that of Padé approximants may yield a more general theory.

The polylogarithm and Hurwitz zeta functions are both special cases of the Lerch transcendental

$$\sum_{n=0}^{\infty} \frac{z^n}{(n+q)^s}$$

It should be possible to extend the techniques in this paper to provide a rapid, globally convergent algorithm for the Lerch transcendental, thus enabling a deeper numerical exploration of its peculiarities.

This paper concludes with the application of the algorithm to render some pretty images (Appendix). Q4

Appendix: Some pretty pictures

Below follow some interesting pictures of the polylogarithm and the Hurwitz zeta function for values of s on the critical line $s = 1/2 + i\tau$. They are shown here, as they are not commonly known. Each figure represents hours of computation on modern computers.

References

1. Abramowitz, M., Stegun, I.A. (eds.): Handbook of mathematical functions, 10th Printing ed. Dover Publications (1972)
2. Apostol, T.M.: Introduction to Analytic Number Theory. Springer, New York (1976)
3. Balanzario, E.P.: A generalized Euler–Maclaurin formula for the hurwitz zeta function. Math. Slovaca **56**(3), 307–316 (2006)
4. Bloch, S.: Structural properties of polylogarithms. Mathematical surveys and monographs, vol. 37, ch. “Introduction to Higher Logarithms”, pp. 275–285. American Mathematical Society (1991)
5. Boas, R.P., Creighton Buck, R.: Polynomial Expansions of Analytic Functions. Academic Press Inc (1964)
6. Borwein, P.: An efficient algorithm for the Riemann zeta function. Constructive experimental and nonlinear analysis, CMS Conference Proceedings 27, pp. 29–34 (preprint, January 1995) (2000) http://www.cecm.sfu.ca/personal/pborwein/PAPERS/P155.pdf
7. Brezinski, C.: Rational approximation to formal power series. J. Approx. Theory **25**, 295–317 (1979)
8. Brezinski, C., Zaglia, M.R.: Extrapolation Methods. North Holland (1991)
9. Costin, O., Garoufalidis, S.: Resurgence of the fractional polylogarithms (2007) arXiv:math. CA/0701743
10. Crandall, R.E.: Note on fast polylogarithm computation. http://people.reed.edu/~crandall/papers/Polylog.pdf (January 2006)
11. Martin, D.W., Miller, G.F., Olver, F.J.W., Clenshaw, C.W., Goodwin, E.T., Wilkinson, J.W. (eds.): Modern computing methods, 2nd edn. Philosophical Library (1961)
12. Kirtman, B., Weniger, E.J.: Extrapolation methods for improving the convergence of oligomer calculations to the infinite chain limit of quasi-onedimensional stereoregular polymers. Comput. Math. Appl. **45**, 189–215 (2003) arXiv:math.NA/0004115
13. Free Software Foundation: Gnu multiple precision arithmetic library. http://www.swox.com/gmp/
14. Graves-Morris, Jr. P.R., Baker, G.A.: Pad<c3><a9> Approximants. Cambridge University Press (1996)
15. Hain, R.M.: Classical polylogarithms (1992) arXiv:alg-geom/9202022
16. Hasse, H.: Ein Summierungsverfahren fur die Riemannsche ζ-Reihe. Math. Z. **32**, 458–464 (1930)
17. Crandall, R.E., Borwein, J.M., Bradley, D.M.: Computational strategies for the Riemann zeta function. J. Comput. Appl. Math. **121**, 247–296 (2000) http://www.maths.ex.ac.uk/~mwatkins/zeta/borwein1.pdf
18. Jonquière, A.: Notes sur la série (polylogarithm). Bull. Soc. Math. France **17**, 142–152 (1889)
19. Karatsuba, E.A.: Fast evaluation of the hurwitz zeta function and Dirichlet L-series. Problem. Peredachi Informat. **34**(4), 342–353 (1998)
20. Lewin, L.: Polylogarithms and Associated Functions. North Holland (1981)
21. Hain, R.M., Macpherson, R.: Structural properties of polylogarithms. Mathematical Surveys and Monographs, vol. 37, ch. “Introduction to Higher Logarithms”, pp. 337–353. American Mathematical Society (1991)
22. Milne-Thomson, A.M.: The Calculus of Finite Differences. Chelsea Publishing (1933)
23. Bateman Manuscript Project: Higher transcendental functions, vol. 1. McGraw Hill, New York (1953)

24. Slezeviciene, R.: An efficient algorithm for computing Dirichlet L-functions. Integral Transforms Spec. Funct. **15**(6), 513–522 (2004)
25. Soff, G., Joachim Weniger, E., Jentschura, U.D., Mohr, P.J.: Convergence acceleration via combined nonlinear-condensation transformations. Comput. Phys. Commun. **116**, 28–54 (1999) arXiv:math.NA/9809111
26. Cohen, H., Villegas, F.R., Zagier, D.: Convergence acceleration of alternating series. Exp. Math. **9**(1), 3–12 (2000) http://www.math.utexas.edu/~villegas/publications/conv-accel.pdf
27. Weniger, E.J.: Nonlinear sequence transformations for the acceleration of convergence and the summation of divergent series. Comput. Phys. Rep. **10**, 189–371 (1989) arXiv math. NA/0306302
28. Weniger, E.J.: Algorithms for approximation, ch. Asymptotic approximations to truncation errors of series representations for special functions, pp. 331–348, Springer-Verlag (2007) arXiv:math/0511074
29. Vetterling, W.T., Press, W.H., Teukolsky, S.A., Flannery, B.P.: Numerical Recipes in C, the Art of Scientific Computing, 2nd edn. Cambidge University Press (1988)
30. Wood, D.: The computation of polylogarithms. Tech. Report 15-92*, University of Kent, Computing Laboratory, University of Kent, Canterbury, UK (June 1992) http://www.cs.kent.ac.uk/pubs/1992/110

AUTHOR QUERIES

AUTHOR PLEASE ANSWER ALL QUERIES

Q1. Affiliation address for author was provided, please check.
Q2. Dummy citations for Figures 5–9, 12 and 15 were inserted. Please check.
Q3. Please check if Fig. 15 was captured correctly.
Q4. A dummy citation for Appendix was inserted. Please check if appropriate.