COMPUTING DEGREE 1 L-FUNCTIONS RIGOROUSLY

David J. Platt

School of Mathematics

September 2011

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Doctor of Philosophy in the Faculty of Science

Wordcount: 15 767

Abstract

We describe a new, rigorous algorithm for efficiently and simultaneously computing many values of the Riemann zeta function on the critical line by exploiting the fast Fourier transform (FFT). We apply this to locating non-trivial zeros of zeta to high precision which are in turn used as input to our own implementation of the Lagarias and Odlyzko analytic algorithm to compute $\pi(x)$, the prime counting function. We confirm the value of $\pi(x)$ for $x = 10^{23}$, matching the largest unconditional result to date.

We then turn to Dirichlet L-functions and detail a version of Booker's rigorous algorithm for generic L-functions, tailored to this application. We employ this for computations with characters of relatively small modulus. For larger modulus, we describe a new algorithm and its implementation. Both again rely on the FFT to compute many values simultaneously and hence achieve efficiency. We use a combination of these two algorithms to extend the work of Rumely and verify the generalised Riemann hypothesis (the GRH) for all characters modulus $q \leq 100\ 000$ to height T such that qT is at least 100 000 000. We then confirm rigorously the non-vanishing of $L_\chi(1/2)$ for all characters of modulus $q \leq 2\ 000\ 000$ before finishing with some comparisons of computed data to predictions from random matrix theory.

Dedication

To Andrea.

Acknowledgements

I would like to thank the administrators, graduate students, post doctoral researchers and academic staff of the department for their contribution to the pleasant and stimulating environment that is number theory at Bristol. In particular, I would like to acknowledge Prof. Trevor Wooley and Dr. Thomas Jordan for their insightful comments at my annual reviews, and Prof. Jon Keating and Dr. Nina Snaith for affording me a glimpse into the exciting world of random matrix theory. This project has required significant computational effort and the IT staff in the department and at the Advanced Computing Resource Centre have provided invaluable assistance in that regard.

I consider myself very fortunate to have been given an opportunity to return to academic life after many years in industry. I am therefore more than grateful to my supervisor, Dr. Andrew Booker, firstly for accepting me as his student and then for identifying an area of research that I continue to find fascinating. Andrew's patient and insightful explanations, his careful and considerate criticism of my bad ideas balanced by his wholehearted support of my better ones, have meant that my motivation to explore has never faltered. I also owe him a great deal for the effort he put in helping me to find a position to allow me to continue to do mathematics.

Finally, I wish to thank my friends and family for their support, and especially Gaynor, my soul-mate. Gaynor has put up with my bad moods when things weren't going as well as I wanted and, worse, my ham fisted attempts to explain each minor success I had along the way. I am a very lucky man.

Author's Declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

David J. Platt

Date: September 2011

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

number theory *Noun*
The study of integers, their properties and the relationship between integers - Collins English Dictionary

Number theory has been part of the mathematical landscape since at least the time of Euclid circa 300 years B.C. (see Books VII and IX of "Elements"). Somewhat later, in 1837, Dirichlet [29] kick started analytic number theory when he used the properties of the L-functions $L_\chi(s)$ that now bear his name to show that there are infinitely many primes in arithmetic progressions $a+bn$ with $(a,b)=1$. In 1859, Riemann published his paper [70] establishing a link between the (Riemann) zeta function, $\zeta(s)$, and the primes via his explicit formula.

Since then, the ability to compute values of these functions, particularly on the critical line $\Re s = 1/2$, has been of practical importance to number theorists, as well as being of genuine academic interest in its own right. In particular, Riemann's guess that all the non-trivial zeros of ζ have real part $1/2$, the Riemann Hypothesis (RH), and its big brother the Generalised Riemann Hypothesis (GRH) which says the same thing for L_χ, have been taxing us for

over 150 years. Riemann himself was the first to try and compute values of ζ on the 1/2 line to test his hypothesis and it is not surprising that Alan Turing used his access to the Manchester Mark I, one of the earliest stored program computers, to do the same.

We will describe new, rigorous and efficient algorithms for both ζ and L_χ and, by way of example, show some applications in which these new algorithms are useful.

1.2 Philosophy

We will be at great pains to convince the reader that the computations we describe are rigorous and we will (presumptively) refer to the results of those computations as theorems. Rigorous computation requires a great deal of effort. We must produce explicit bounds for every approximation, replacing "big oh" with a number in every case. We must also manage the errors introduced by having to represent real numbers by machine floating point approximations and the rounding errors that occur when operations are performed on these floats.

Many researchers do not feel the need to go to these lengths. Some argue that since hardware, operating systems and compilers are all man made, rigorous computation is an oxymoron. Others rely on the random walk cancellation of rounding errors, arguing that statistically the doomsday scenario of catastrophic accumulation is so unlikely it can be ignored. We beg to differ on the following grounds:

- It is hard to make the necessary bounds explicit, but it is possible. Using, for example, interval arithmetic to manage the problems inherent in floating point does cost CPU cycles, but that simply means rigorous computations need to be a bit less ambitious.

- Arguing that someone else (hardware designer, compiler writer etc.) may

have made a mistake seems a very poor justification for not even trying to get our bit right.

- Hardware and software bugs are not as insidious as some might have us believe. Those that are known are no threat, we simply tiptoe round them. The last major hardware bug (Intel's Pentium fdiv debacle) did not stay hidden for long before Nicely uncovered it computing Brun's sum [22]. (As an aside, Intel were actually already aware of it!) Thorough testing, porting code to different platforms and using different compilers and operating systems all drastically reduce exposure to this threat and constitute good practice.

- Rigorous computation is an academic challenge in its own right. We pure mathematicians should be very careful before we label anything a waste of time and effort, lest we all get tarred with the same brush.

1.3 Structure of this Document

Immediately after this introduction, we will dispense with some mathematical and computational background that will be needed later. In particular, we will define ζ and L_χ and point out some of their properties that we will later exploit. On the computational side, we will discuss how to implement interval arithmetic and also describe the Discrete Fourier Transform, the efficient implementation of which is at the core of the algorithms we have developed.

We then survey existing methods for computing ζ and L_χ, before moving on to the first new algorithm, that for computing ζ on the half line. The next section describes the application of this algorithm to an analytic version of the prime counting function.

Moving on to L_χ, we describe two algorithms, one a specialisation of an existing method due to Booker for generic L-functions and the other a new algorithm of our design. This time we describe three applications of these

algorithms, one investigating the Generalised Riemann Hypothesis (GRH), one examining the non-vanishing of $L_\chi(1/2)$ and the third comparing statistics for $L_\chi(1/2)$ for χ of large modulus with predictions from Random Matrix Theory (RMT).

We then finish with a brief discussion of areas for further research based on the ideas explored herein.

1.4 Notation

Although we believe most of our notation to be standard, we list below the conventions adopted herein.

$\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$	the resp. integers, rationals, reals, complex plane
$\lfloor t \rfloor$, $\{t\}$	the resp. integer part, fractional part of t, $t \in \mathbb{R}_{\geq 0}$
$\zeta(s, \alpha)$	the Hurwitz Zeta Function (see section 2.2)
$\zeta(s)$	the Riemann Zeta Function (see section 2.3)
$B_n(t)$	the n'th Bernoulli polynomial in t
$B_n := B_n(0)$	the n'th Bernoulli number
$\Re s$, $\Im s$, $\overline{s}$	the resp, real part, imaginary part, complex conjugate of s
$\Gamma(s)$	the gamma function
$\mathrm{e}(x) := \exp(2\pi i x)$	the complex exponential
$F(x) := \int_{-\infty}^{\infty} f(t)\,\mathrm{e}(-tx)\,\mathrm{d}t$	the Fourier transform of f
$f(t) := \int_{-\infty}^{\infty} F(x)\,\mathrm{e}(tx)\,\mathrm{d}x$	the inverse Fourier transform of F
$\mathrm{Ei}(x)$	the exponential integral, defined for $x > 0$ by $\int_x^{\infty} \frac{\exp(-t)}{t}\,\mathrm{d}t$ and analytically continued to $\mathbb{C} \setminus \mathbb{R}_{\leq 0}$
$\varphi(n)$	Euler's totient function

In addition, given complex valued functions f and g, we write $f(z) = \mathcal{O}(g(z))$ (or equivalently $f(z) \ll g(z)$ or $g(z) \gg f(z)$), to mean that there is a constant $C > 0$ such that $|f(z)| < C\,|g(z)|$ over some domain that should be clear from the context.

Chapter 2

Mathematical Prerequisites

2.1 L-Functions and the Selberg Class

In [77], Selberg introduced the (now eponymous) class of functions, $\mathbb{S}$, and made several important conjectures regarding it. A function F defined for $\Re s > 1$ by a Dirichlet series

$$F(s) = \sum_{n=1}^{\infty} \frac{a_n}{n^s}$$

is a member of $\mathbb{S}$ if it satisfies the following 4 axioms:

1. Analyticity: there exists an $m \in \mathbb{Z}_{\geq 0}$ such that $(s-1)^m F(s)$ is entire and of finite order.

2. Ramanujan hypothesis: $a_n \ll_\epsilon n^\epsilon$ for any fixed $\epsilon > 0$.

3. Functional equation: there exists a function $\gamma_F(s)$ of the form

$$\gamma_F(s) = \omega Q^s \prod_{j=1}^{k} \Gamma(w_j s + \mu_j)$$

with $k \in \mathbb{Z}_{\geq 0}$, $|\omega| = 1$, $Q, w_j > 0$, $\Re \mu_j \geq 0$ and such that, if we define $\Phi(s) := \gamma_F(s) F(s)$, then

$$\Phi(s) = \overline{\Phi(\overline{1-s})}.$$

We will refer to ω as the root number.

4. Euler product: $a_1 = 1$ and $\log F(s) = \sum_{n=1}^{\infty} \frac{b_n}{n^s}$ with $b_n = 0$ unless n is a prime power and $b_n \ll n^\theta$ for some $\theta < 1/2$.

We define the degree of $F \in \mathbb{S}$ to be

$$d_F = 2\sum_{j=1}^{k} w_j$$

and write $\mathbb{S}(d)$ for the subset of $\mathbb{S}$ containing functions with $d_F = d$. We note that the only function in $\mathbb{S}(0)$ is $F(s) = 1$ and that $\mathbb{S}(d)$ is empty for $0 < d < 1$ [25]. The structure of $\mathbb{S}(1)$ was settled by Kaczorowski and Perelli [42] who showed that it contains precisely the Dirichlet L-functions, with arbitrary imaginary displacement, and Riemann's zeta function. We will describe these functions shortly.

2.2 The Hurwitz Zeta Function

(See for example [3])

The Hurwitz Zeta function $\zeta(s, \alpha)$ is defined initially for $\Re s > 1$ and $\alpha \in (0, 1]$ by

$$\zeta(s, \alpha) := \sum_{n=0}^{\infty} \frac{1}{(n + \alpha)^s}.$$

It has analytic continuation to the entire complex plane with the exception of a simple pole with residue 1 at $s = 1$ and we have the following identity

$$\zeta\left(s, \frac{1}{2}\right) = (2^s - 1)\zeta(s, 1).$$

2.3 Riemann's Zeta Function

(See, for example [83])

Riemann's zeta function is defined initially for $\Re s > 1$ by

$$\zeta(s) := \sum_{n=1}^{\infty} \frac{1}{n^s}.$$

It has analytic continuation to the entire complex plane with the exception of a simple pole at $s = 1$ with residue 1 and we have the following functional equation

$$\pi^{-\frac{s}{2}}\Gamma\left(\frac{s}{2}\right)\zeta(s) = \pi^{-\frac{1-s}{2}}\Gamma\left(\frac{1-s}{2}\right)\zeta(1-s).$$

Since neither ζ nor Γ have poles to the right of $s = 1$, the simple poles of $\Gamma\left(\frac{s}{2}\right)$ for $s \in \{-2, -4, \ldots\}$ must correspond to simple zeros of ζ. In addition to these "trivial" zeros, ζ has an infinite number of zeros with real part in the interval $(0, 1)$. The conjecture that all of these non-trivial zeros have real part exactly $\frac{1}{2}$ is the Riemann Hypothesis (RH).

If we define

$$\Lambda(t) := \pi^{-\frac{it}{2}}\Gamma\left(\frac{\frac{1}{2}+it}{2}\right)\exp\left(\frac{\pi t}{4}\right)\zeta\left(\frac{1}{2}+it\right) \tag{2.3.1}$$

then we observe that $\Lambda(t)$ has the same zeros as $\zeta\left(\frac{1}{2}+it\right)$ and, by the functional equation, is real valued. The exponential factor is designed to counteract the decay of the gamma factor as t increases.

The Riemann Zeta function can be written in terms of the Hurwitz Zeta function simply by

$$\zeta(s) = \zeta(s, 1).$$

We have $\zeta \in \mathbb{S}(1)$ with $m = 1$, $\omega = 1$, $Q = \pi^{-1/2}$, $k = 1$, $w_1 = 1/2$ and $\mu_1 = 0$.

2.4 Dirichlet L-Functions

2.4.1 Dirichlet Characters

(See, for example [26])

A Dirichlet character χ of modulus $q \in \mathbb{Z}_{>0}$ is a q periodic, completely multiplicative arithmetic function such that $\chi(1) = 1$ and if $(n, q) \neq 1$ then $\chi(n) = 0$.

Thus for $(n, q) = 1$ we have that $\chi(n)$ is a root of unity.

When $q = 1$ we have the trivial character $\chi(n) = 1$.

For a given q there are $\varphi(q)$ distinct characters. The character which takes the value 1 for all n co-prime to q is known as the principal character χ_0 (modulo q).

If we can write $\chi = \chi_0\chi'$ where χ' has modulus less than that of χ, then χ is referred to as an imprimitive character, otherwise it is primitive.

2.4.2 Forming Dirichlet L-Functions

(See, for example [26])

For $\Re s > 1$ and χ a Dirichlet character modulus q, we define the Dirichlet L-function $L_\chi(s)$ by

$$L_\chi(s) := \sum_{n=1}^{\infty} \frac{\chi(n)}{n^s}.$$

In the case $q = 1$ we have $L_\chi(s) = \zeta(s)$.

Dirichlet L-functions have analytic continuation. With principal characters, there is a simple pole at $s = 1$ with residue

$$\prod_{\substack{p \text{ prime} \\ p|q}} \left(1 - \frac{1}{p}\right).$$

For non-principal characters the analytic continuation of $L_\chi(s)$ is entire.

Dirichlet L-functions formed from primitive characters satisfy a functional equation. First, we define a_χ, the parity of a Dirichlet character, by

$$a_\chi := \frac{1 - \chi(-1)}{2}$$

so if $\chi(-1) = 1$ we have $a_\chi = 0$ (an even character) and if $\chi(-1) = -1$ we have $a_\chi = 1$ (an odd character).

Now we define the Gaussian sum $\tau(\chi)$ by

$$\tau(\chi) := \sum_{n=1}^{q} \chi(n)\, \mathrm{e}\left(\frac{n}{q}\right),$$

the root number by

$$\omega_\chi := i^{a_\chi}\sqrt{q}(\tau(\chi))^{-1}$$

and the function ξ_χ by

$$\xi_\chi(s) := \left(\frac{\pi}{q}\right)^{-\frac{s+a_\chi}{2}} \Gamma\left(\frac{s+a_\chi}{2}\right) L_\chi(s).$$

Then the functional equation can be written

$$\xi_{\overline{\chi}}(1-s) = \omega_\chi \, \xi_\chi(s).$$

Thus, by a similar argument to that applied to ζ, we see for even primitive characters of modulus > 2, the function $L_\chi(s)$ must have trivial zeros at $s \in \{0, -2, -4, \ldots\}$ whereas for odd characters these zeros are at $s \in \{-1, -3, \ldots\}$. Dirichlet L-functions of both even and odd primitive characters have an infinite number of non-trivial zeros with real part in the interval $(0, 1)$. The Generalised Riemann Hypothesis (GRH) is the conjecture that all these non-trivial zeros have real part exactly $\frac{1}{2}$.

For a primitive character χ, we set $\epsilon_\chi = \omega_\chi^{1/2}$ such that $\mathrm{Arg}(\epsilon_\chi) \in (-\pi/2, \pi/2]$ and then define $\Lambda_\chi(t)$ by

$$\Lambda_\chi(t) := \epsilon_\chi \left(\frac{q}{\pi}\right)^{\frac{it}{2}} \Gamma\left(\frac{\frac{1}{2} + a_\chi + it}{2}\right) \exp\left(\frac{\pi t}{4}\right) L_\chi\left(\frac{1}{2} + it\right).$$

Now $\Lambda_\chi(t)$ has the same zeros as $L_\chi\left(\frac{1}{2} + it\right)$ and, by the functional equation, is real valued. The exponential factor is designed to counteract the decay of the gamma factor as t increases.

Dirichlet L-functions can be written in terms of the Hurwitz Zeta function, except at $s = 1$ using

$$L_\chi(s) = q^{-s} \sum_{n=1}^{q} \chi(n) \zeta\left(s, \frac{n}{q}\right). \tag{2.4.1}$$

For primitive χ we have $L_\chi \in \mathbb{S}(1)$ with $m = 0$, $\omega_\chi = i^{a_\chi}\sqrt{q}(\tau(\chi))^{-1}$, $Q = \sqrt{\frac{q}{\pi}}$, $k = 1$ and $w_1 = 1/2$. In the case of even χ we have $\mu_1 = 0$, for odd χ we have $\mu_1 = 1/2$.

2.5 Turing's Method

2.5.1 Turing's Method and ζ

In 1953, a year before his death, the London Mathematical Society published an account by Alan Turing of his investigations into ζ and the Riemann Hypothesis using the Manchester Mark I computer [85]. In that paper, Turing described his method for verifying RH for a segment of the critical line. This boiled down to determining the number of zeros of ζ in a rectangle including the segment of the critical line under test, and comparing this with the number of zeros actually found by computing values of Λ and looking for changes in sign. If the two agree, RH is shown to hold for that piece of the critical line.

Theorem 2.5.1. *For t not the ordinate of a zero nor a pole of ζ, define*

$$S(t) := \frac{1}{\pi}\Im \int_{\infty}^{\frac{1}{2}} \frac{\zeta'}{\zeta}(\sigma + it)\,\mathrm{d}\sigma,$$

and when t is a zero or a pole, define

$$S(t) := \lim_{\epsilon \to 0+} S(t+\epsilon).$$

Now for t not the ordinate of a zero of ζ, define $N(t)$ to be the number of zeros of $\zeta(s)$ with $\Re s \in (0,1)$ and $\Im s \in [0,t]$. Then

$$N(t) = \frac{1}{\pi}\left[\Im \log \Gamma\left(\frac{\frac{1}{2}+it}{2}\right) - \frac{t\log\pi}{2}\right] + 1 + S(t). \tag{2.5.1}$$

Proof. See page 128 of Edwards [30]. □

Theorem 2.5.2. *For $T > 168\pi$ and $h > 0$ we have*

$$\left|\int_{T}^{T+h} S(t)\,\mathrm{d}t\right| \leq 2.3 + 0.128\log(T+h).$$

Proof. This is the main result of Turing's 1953 paper. □

We note that the constants 2.3 and 0.128 were subsequently improved to 1.7 and 0.114 respectively by Lehman [52]. Recently, Trudgian has supplied the improved constants 2.067 and 0.059, this time optimised for T around $2\pi \times 10^{12}$ (Theorem 2.2 of [84]).

Turing's method now proceeds by taking the integral of Equation 2.5.1 over a small segment of the critical line from T to $T + h$. If the assumed number of zeros up to T is incorrect, it will quickly become apparent through a contradiction of Theorem 2.5.2 (or one of its later improvements). This process is made absolutely rigorous in [13].

2.5.2 Turing's Method and Dirichlet L-Functions

Theorem 2.5.3. *Given $t_0, h > 0$ such that neither t_0 nor $t_0 + h$ is the imaginary part of a zero of $L_\chi(s)$, let $N_\chi(t_0)$ be the number of zeros, counted with multiplicity, of $L_\chi(s)$ with $|\Im(s)| \leq t_0$ and $\Re(s) \in (0,1)$. Let $\widetilde{N}_{t_0,\chi}(t)$ count the zeros of $L_\chi(s)$ with $\Im(s) \in [t_0, t]$, starting at 0 at t_0 and increasing by 1 at every zero.*

Now for t not the ordinate of a zero of L, define $S_\chi(t)$ by

$$S_\chi(t) := \frac{1}{\pi} \Im \int_\infty^{\frac{1}{2}} \frac{L'_\chi}{L_\chi}(\sigma + it)\, \mathrm{d}\sigma$$

and like $S(t)$, take $S_\chi(t)$ to be upper semi-continuous. Then we have

$$\begin{aligned} N_\chi(t_0) =& \frac{1}{h\pi} \left[2h + \frac{2ht_0 + h^2}{2} \log\left(\frac{q}{\pi}\right) + 2 \int_{t_0}^{t_0+h} \Im \log \Gamma \left(\frac{1/2 + a_\chi + it}{2} \right) dt \right. \\ & \left. - \int_{t_0}^{t_0+h} \widetilde{N}_{t_0,\chi}(t)\, \mathrm{d}t - \int_{t_0}^{t_0+h} \widetilde{N}_{t_0,\overline{\chi}}(t) dt + \int_{t_0}^{t_0+h} S_\chi(t)\, \mathrm{d}t + \int_{t_0}^{t_0+h} S_{\overline{\chi}}(t)\, \mathrm{d}t \right]. \end{aligned}$$

Proof. We start with Equation 4-2 of [13] and specialise to Dirichlet L-functions. We treat conjugate characters in pairs to avoid problems with the arbitrary choice of the square root of ω_χ and to allow for the possibility that $S_\chi(0)$ isn't small. Finally, we integrate both sides from t_0 to $t_0 + h$. □

Rumely extended Turing's method to Dirichlet L-functions.

Theorem 2.5.4. *(Rumely). For $T > 50$ and $h > 0$*

$$\left| \int_T^{T+h} S_\chi(t)\,\mathrm{d}t \right| \leq 1.8397 + 0.1242 \log\left(\frac{q(T+h)}{2\pi}\right).$$

Proof. Theorem 2 of [76]. □

In a personal communication, Trudgian has provided revised constants optimised for qT in the region of 10^8. These are 2.17618 and 0.0679955 respectively.

Applying Turing's method to Dirichlet L-functions is now identical to that for ζ, except for the pairing of conjugates.

2.6 Rigorous Up-sampling

We aim to compute $\zeta(s)$ and $L_\chi(s)$ on a relatively coarse lattice and to use these values to interpolate intermediate ones to the necessary precision, for example to distinguish zeros on the critical line. We start with a result from signal processing theory.

Theorem 2.6.1. *(Whittaker-Shannon Sampling Theorem) Let $f(t)$ be a continuous, real valued function with Fourier Transform $F(x)$ such that $F(x) = 0$ for $|x| > B > 0$ (i.e. $f(t)$ is band-limited with bandwidth B). Also, define*

$$\mathrm{sinc}(x) := \frac{\sin(x)}{x}.$$

Then

$$f(t) = \sum_{n\in\mathbb{Z}} f\left(\frac{n}{2B}\right) \mathrm{sinc}\left(2B\pi\left(\frac{n}{2B} - t\right)\right),$$

when this sum converges.

Proof. See [86]. □

To make this process rigorous, we need to examine two sources of error

- the error introduced by truncating the sum
- the error introduced if the function is only approximately band-limited

The former will be dealt with on a case by case basis. The latter, referred to as aliasing in signal processing circles, is the subject of a theorem due to Weiss.

Theorem 2.6.2. *Let $f(t)$ be a real valued function with Fourier Transform $F(x)$ such that*

1. $\int_{-\infty}^{\infty} |F(x)|dx < \infty$
2. *$F(x)$ is of bounded variation on $\mathbb{R}$*
3. *when F has a jump discontinuity at x, then $F(x) = \lim_{\epsilon \to 0^+} \frac{F(x-\epsilon)+F(x+\epsilon)}{2}$.*

Then

$$\left| f(t) - \sum_{n \in \mathbb{Z}} f\left(\frac{n}{2B}\right) \operatorname{sinc}\left(2B\pi\left(t - \frac{n}{2B}\right)\right) \right| \leq 4 \int_{B}^{\infty} |F(x)|\, dx.$$

Proof. For a full proof, see for example [16]. Less formally, we consider the Dirac Delta function $\delta(t)$ and define the Dirac Comb function $\Delta_w(t)$ by

$$\Delta_w(t) := \sum_{n \in \mathbb{Z}} \delta(t - wn).$$

Thus our sampled function is given by

$$f_{sam}(t) = f(t) \times \Delta_{1/(2B)}(t)$$

and its Fourier transform by

$$\begin{aligned} F_{sam}(x) &= \int_{-\infty}^{\infty} f(t) \times \Delta_{1/(2B)}(t)\, \mathrm{e}(-xt)\, \mathrm{d}t \\ &= F(x) * 2B\Delta_{2B}(x) \\ &= 2B \sum_{n \in \mathbb{Z}} F(x + 2nB). \end{aligned}$$

Thus the effect of convolving with a comb is to create multiple copies of the original function with centres $2B$ apart.

Our next step is to multiply by a rectangle function of width $2B$ and height $1/2B$. However, because $f(t)$ is not band limited, the multiple copies of $F_{sam}(x)$ "leak" into the frequencies selected by the rectangle function. To quantify this error, we consider just those copies to the left of the central copy $F(x)$. The first is $2BF(x+2B)$ and the total error introduced into $[-B, B]$ by this copy is $\leq \int_B^{3B} |F(x)| \, \mathrm{d}x$. The next copy is $2BF(x+4B)$ and the absolute value of the error introduced is $\leq \int_{3B}^{5B} |F(x)| \, \mathrm{d}x$. Continuing, the total error introduced by all the copies to the left of centre is bounded in absolute value by

$$\sum_{n \in \mathbb{Z}_{>0}} \int_{(2n-1)B}^{(2n+1)B} |F(x)| \, \mathrm{d}x = \int_B^{\infty} |F(x)| \, \mathrm{d}x.$$

Now since $W(t)$ is real valued, the spectrum to the right of the central copy is simply the complex conjugate of that to the left, so to get the total error from copies to the left and right, it suffices to double this bound. Finally, we double again to take account of the spectrum of the central copy beyond $[-B, B]$ that is cut off by the rectangle filter and we have the factor of 4 in the theorem. □

We note that this theorem can obviously be used to up-sample at specific points of interest. In addition, if we take the DFT of a band limited function, we can pad the result with zeros (or, to be rigorous, with a small error term) and take the inverse DFT. Thus if we require n equally spaced, interpolated values we can achieve this in $\mathcal{O}(n \log n)$ operations or, on average, $\mathcal{O}(n^{\epsilon})$ per value.

Chapter 3

Computational Prerequisites

3.1 Interval Arithmetic

3.1.1 History and Background

"A digital computation is a finite sequence of inexact arithmetic operations" - R.E. Moore [57]

This rather pessimistic quote by Ramon Moore is the first line of one of the papers that marked a renewed interest in interval arithmetic spurred by the advance of the digital computer. Two general problems arise:

- the finite nature of computer memory makes it incapable of exactly representing almost all real numbers, and
- our desire to obtain results in finite time requires that infinite sums and products be truncated.

Many techniques have been developed by numerical analysts to manage such sources of error. One way to try to circumvent the problems of finite representation is run a computation several times with different precisions (i.e. using more or less memory to represent each number). An example quoted in [58] is of computing

$$333.75b^6 + a^2(11a^2b^2 - b^6 - 121b^4 - 2) + 5.5b^8 + \frac{a}{2b}$$

with $a = 77617.0$ and $b = 33096.0$. Computing powers by repeated multiplication on an IBM 370 (those were the days) using single (about 7 decimal digits), double (about 16) and extended (about 33) precision gives respectively

$$1.17260361\ldots$$

$$\underline{1.17260}394005317847\ldots$$

$$\underline{1.17260394005317}63185\ldots$$

where the underlined digits are those we might be induced to trust. The exact result is $-0.827396\ldots$! Moore then goes on to state "However, it is often prohibitively difficult to tell in advance of a computation how many places must be carried to guarantee results of required accuracy."

Interval arithmetic is one method of mechanising this "prohibitively difficult" task, at the expense of some computational efficiency. Instead of representing a real number as the nearest representable machine number, we store two machine numbers representing an interval known to contain the target real. We then define the basic operations $+ - \times \div$ for intervals (languages such as C++ which allow operator overloading are useful in this respect). These, together with the functions sin, cos, exp, log and atan, provide a sufficiently rich basis.

With such an arithmetic to hand, the management of errors caused by the representational limitations of any chosen precision comes for free. With a little more work, the programmer can explicitly and rigorously manage errors from truncating series. If a bound for the error is known a priori, it can simply be added as an interval $[-\epsilon, \epsilon]$ to the result of the truncated calculation. Indeed, interval arithmetic is a useful tool in generating such bounds, for example using the Lagrange form for the remainder of a Taylor series.

Interval arithmetic also provides a useful belt to the numerical analyst's braces. Often the rounding errors that build up during a complex calculation depend on the order in which the sub calculations are performed. Using scalar arithmetic, there is no warning of a calamitous choice, just complete nonsense

at the end. Even if we are lucky enough to recognise that it is complete nonsense, the problem of determining where the logic error has occurred is not trivial. Using interval arithmetic, however, provides a smoking gun. At some point in the computation, the intervals will cease abruptly to be nice and narrow and that is where to start looking.

Of course, all of this extra functionality costs. In terms of memory, intervals take up twice the space of their scalar equivalents. The loss in efficiency is twofold. First, computing the sum or difference of two intervals is twice as expensive as the scalar equivalent and multiply can be four times worse. Secondly, modern CPU's attempt to exploit pipe-lining of instructions to achieve improved throughput. Changing rounding mode (to round down when computing the left end point of an interval, then to round up to compute the right end point) destroys the pipeline with a consequent loss in performance.

Whilst the former efficiency issue is inherent to interval arithmetic, with clever data structures and algorithms, the switching of rounding modes can be all but eliminated.

3.1.2 The int_double Class

Following the work of Lambov [51], we have written a C++ class int_double to implement double precision interval arithmetic. It stores intervals as two 64 bit IEEE 754 [39] double precision floating points in contiguous memory aligned to a 16 byte boundary representing the left and right endpoints of a real interval. The operators $+,-,*$ and / are overloaded to take int_double arguments and integers and doubles are coerced to int_doubles as required.

A function *sqr* is provided to compute the square of an interval. Doing this the naive way by multiplication, e.g. $[-1,2]^2 = [-1,2] * [-1,2] = [-2,4]$ is clearly suboptimal as the correct answer is $[0,4]$. This is an example of a general phenomenon first identified by Moore [58], in that operations where the two or more operands are effectively the same interval are problematic.

Other simple cases include $x - x$ and $x \div x$.

This functionality is realised using in-line assembler making use of the 128 bit XMM registers and SSE instructions available on most modern Intel processors [40]. By storing the right hand endpoint in negated form and setting the default rounding mode of SSE instructions to round towards $-\infty$, we can achieve + by simply adding left to left and right to right (a single SSE instruction), unary − by swapping the left and right endpoints (one SSE instruction), binary − by swapping followed by addition (two SSE instructions). Multiplication takes 21 instructions to form the 4 possible multiplications correctly rounded and select the lowest and highest results. Division requires 15 instructions.

In addition to setting the rounding mode for SSE instructions, we also clear the Flush to Zero and Denormals are Zero flags, neither of which are IEEE 754 compliant.

The square root of an interval is calculated using the floating point unit (whose rounding mode is round to nearest). IEEE compliance guarantees that the result of a square root is within one unit of last place (ulp) of the correct result so we expand the resulting interval by one ulp in each direction.

Unfortunately, the same technique cannot be used for exp, log, sin, cos and atan as these lie outwith IEEE 754 and there are no guarantees of accuracy.

Even over reduced ranges of arguments, the Table Maker's Dilemma (first coined by Kahan [43]) makes accurate computation of transcendental functions problematic. This dilemma refers to the problem that to round correctly to fixed precision (whether rounding to nearest, or to $\pm\infty$) we may need to know the result to a much higher precision. For example (in binary), say our fixed precision is 2 places after the decimal point then and we want the nearest 2 bit representable number to the true result. Computing $f(x)$ to 2 places might give us a result of 0.10 which means $f(x) \in [0.01, 0.11]$. If we compute with an extra bit, say we get 0.100. This means $f(x) \in [0.011, 0.101]$ so we still don't know what the first two bits should be. In pathological cases we might need

many more bits before we can resolve to our target precision.

We use Muller and de Dinechin's "Correctly Rounded Mathematical Library" [59] to compute these elementary transcendental functions in software. These library routines are based on a rigorous determination of how many extra bits of precision are required to eliminate the Table Maker's Dilemma in double precision (more than 100 in some cases).

3.1.3 The int_complex Class

Extending real interval arithmetic to handle complex intervals is a non-trivial problem. Many approaches have been proposed and tried, including using intervals represented by rectangles, circles and sectors to name but three [34] [47] [54] [65] [72].

All these representations suffer from not being closed under the basic operations addition, subtraction, multiplication and inversion. For example, a rectangle multiplied by a rectangle results in a region that is not in general rectangular. Rather it will be the convex hull of the (up to) 16 points formed by point-wise multiplication of the corners of the multiplicands. In summary [47]

Representation	Closed Under		
	$Z_1 \pm Z_2$	$Z_1 \times Z_2$	Z^{-1}
Circle	Yes	No	Yes
Rectangle	Yes	No	No
Sector	No	Yes	Yes

The second issue is when an operation fails to be closed for a given representation, how easy is it to obtain the smallest possible rectangle, circle or sector respectively that just includes the necessary subset of the complex plane.

In the case of rectangles and multiplication, the obvious method produces such a minimal result. However with inversion, computing $\frac{\overline{Z}}{Z\overline{Z}}$ produces too large a result. Rokne and Lancaster describe a variant of inversion that does

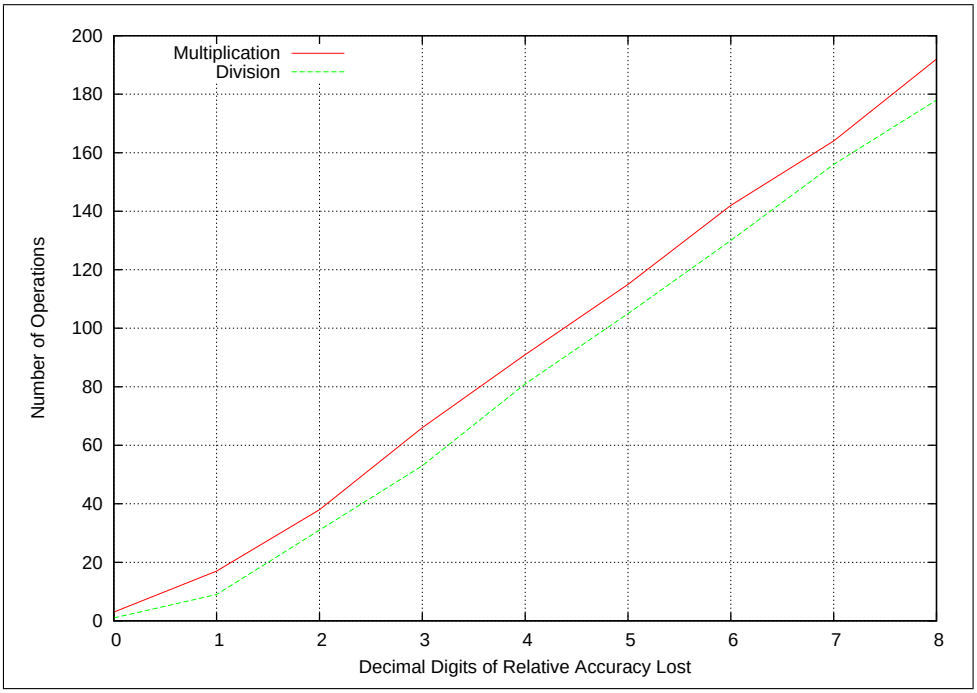

Figure 3.1: The Degradation of Rectangular Complex Interval Accuracy

produce a minimal rectangle [72] but it requires significantly more computation.

Circular representations fail to be closed only under multiplication [34] but a reasonable approximation to the minimal circle formed from $Z_1 \times Z_2$ with Z_1, Z_2 circular with centres c_1, c_2 and radii r_1, r_2 respectively can be obtained by the circle centred at $c_1 \times c_2$ with radius $|Z_1|r_2 + |Z_2|r_1 + r_1 r_2$. Again, this is quite computationally involved and complicated by the fact that in general $c_1 \times c_2$ etc. will result in a complex interval, the radius of which must be added to the final result.

Representations based on sectors fail to be closed under addition and subtraction. Klatte and Ullrich [47] investigated various solutions, the best of which was to enclose the target sectors in circles, add them and then revert to sectors. This is again computationally intensive.

In practical terms, starting with the minimum positive width double precision interval (accurate to 14 or 15 decimal places) repeated multiplication by a similar width interval behaves as expected, in that it takes, for example,

1 000 000 multiplies to lose 6 decimal places of relative accuracy (assuming we avoid over/underflow). Figure 3.1 shows how much worse the situation is with double precision complex intervals implemented as rectangles. We started with the exact point $1 + 0i$ and repeatedly multiplied or divided by a rectangle that straddled the unit circle with a width of 1 ulp in both the real and imaginary component. As expected, division is worse than multiplication, but both are much worse than the real interval case.

Despite these reservations, we decided to adopt rectangular intervals based on the obvious algorithms for reasons of computational efficiency. The needless loss of precision due to the non closed nature of multiplication and inversion, and due to the suboptimal implementation of inversion can be managed in multiple precision architectures by simply starting with more bits to compensate and the loss of efficiency implied by this extra precision might still be less than that gained by the simplicity of the implementation. Time constraints prevented us from investigating other options further.

We wrote a C++ class using two real intervals to represent the real and imaginary parts, and overloaded the four basic operators using in-line assembler. As an aside, we used the high school 4 multiplication method for complex multiplication in favour of the 3 multiplication alternative as the latter produced wider intervals.

The functions for conjugation, norm, modulus, exponentiation, square root, argument and logarithm where implemented using the obvious algorithms in in-line assembler or C++ as appropriate. Where it made sense, these functions were overloaded to take real interval and scalar arguments.

3.1.4 Multiple Precision Interval Arithmetic

For applications that require more precision than the 53 bits (about $14 - 15$ decimal digits) of IEEE-754 doubles, we use Revol and Rouillier's MPFI package [68] [69]. This is written in C and we have extended it in the obvious

way to implement complex arithmetic using rectangular intervals.

3.1.5 The $exp(it)$ Problem

One of the recurring problems in maintaining precision performing rigorous computations with complex numbers is that of computing $\exp(it)$ for t real and $|t|$ large. The periodicity of the exponential function means that we effectively reduce t modulo 2π and thus lose precision in the argument (and hence the result) at the same time. For example, if t is in the region of 6 000 000 this argument reduction will throw away about 6 decimal places. This is particularly an issue if working in double precision intervals as simply throwing more bits at the problem is not an option.

This problem arises in the current context when computing π^{-it} and q^{it} for $q \in \mathbb{Z}_{>1}$. Fortunately, because our algorithms are discrete in t, it is possible to pre-compute a database of values in high precision (using MPFI). Specifically, if we wish to compute q^{it} for $q \in [2, \ldots, Q]$ and $t = \delta T$ for some step size $\delta > 0$ and $T \in [0, \ldots, 2^m - 1]$ we need compute and store the $m(Q-1)$ values of $q^{i\delta 2^k}$ for $k \in [0, \ldots, m-1]$. Reconstructing $q^{i\delta T}$ now reduces to taking the binary expansion of T and taking the product of the relevant precomputed values. The π^{it} case can be achieved with a further m pre-computations.

3.2 The Discrete Fourier Transform

3.2.1 Definition

Given $N \in \mathbb{Z}_{>0}$ complex values denoted X_0 through X_{N-1}, the forward Discrete Fourier Transform (DFT) results in N new values Y_0 through Y_{N-1} where

$$Y_m = \sum_{n=0}^{N-1} X_n \, \mathrm{e}\left(\frac{-nm}{N}\right). \tag{3.2.1}$$

The backward or inverse DFT (iDFT) results from changing the sign in the complex exponential. Performing a forward then backward DFT (or vice

versa) multiplies each datum by N.

3.2.2 Poisson Summation and the DFT

Theorem 3.2.1. *(Poisson Summation Formula) Let f be a function in the Schwartz space with Fourier transform F. Then for $T > 0$*

$$\sum_{n\in\mathbb{Z}} f\left(t+nT\right) = \frac{1}{T}\sum_{n\in\mathbb{Z}} F\left(\frac{n}{T}\right) \mathrm{e}\left(\frac{nt}{T}\right).$$

Furthermore both sides converge uniformly and absolutely to the same limit.

Proof. See, for example [48]. □

Theorem 3.2.2. *Let f be a function in the Schwartz space with Fourier transform F. Let $N = AB$ with $A, B > 0$ and define $\tilde{f}(n) := \sum_{l\in\mathbb{Z}} f\left(\frac{n}{A} + lB\right)$ and $\tilde{F}(m) := \sum_{l\in\mathbb{Z}} F\left(\frac{m}{B} + lA\right)$. Then, up to a constant factor, $\tilde{f}$ and $\tilde{F}$ form a DFT pair of length N.*

Proof. By Poisson summation we have

$$\begin{aligned}\sum_{l\in\mathbb{Z}} f(t+lB) &= \frac{1}{B}\sum_{l\in\mathbb{Z}} F\left(\frac{l}{B}\right) \mathrm{e}\left(\frac{lt}{B}\right)\\ \tilde{f}(n) &= \frac{1}{B}\sum_{l\in\mathbb{Z}} F\left(\frac{l}{B}\right) \mathrm{e}\left(\frac{ln}{N}\right).\end{aligned}$$

We now write $l = l'N + m$ to get

$$\begin{aligned}\tilde{f}(n) &= \frac{1}{B}\sum_{m=0}^{N-1}\sum_{l'\in\mathbb{Z}} F\left(\frac{l'N+m}{B}\right) \mathrm{e}\left(\frac{(l'N+m)n}{N}\right)\\ &= \frac{1}{B}\sum_{m=0}^{N-1} \mathrm{e}\left(\frac{mn}{N}\right)\tilde{F}(m).\end{aligned}$$

This is by definition an iDFT. □

The utility of this theorem will be apparent when f and F both decay quickly enough to allow $\tilde{f}(n)$ and $\tilde{F}(m)$ to be approximated by $f\left(\frac{n}{A}\right)$ and $F\left(\frac{m}{B}\right)$ respectively.

3.2.3 Fast Fourier Transform Algorithms

If we were to rely on the obvious algorithm requiring $\mathcal{O}(N^2)$ operations to compute a length N DFT then we suspect they would have little practical application. It is the existence of fast $\mathcal{O}(N \log N)$ algorithms that makes their use so ubiquitous. Many such Fast Fourier Transform (FFT) algorithms have been developed but here we describe one for lengths which are a power of 2 and later we will discuss Bluestein's algorithm applied to arbitrary lengths. Both are described in more detail in, for example, [15].

3.2.3.1 The Decimation in Time FFT

If we start with a vector X of even length N, we can decompose its DFT as follows:

Theorem 3.2.3. *Let X be a complex valued vector of even length N, let V be the length $N/2$ DFT of its even numbered elements and let W be the DFT of its odds. Then Y, the DFT of X is given by*

$$Y_m = V_{m \bmod N/2} + \mathrm{e}\left(-\frac{m}{N}\right) W_{m \bmod N/2}.$$

Proof. We start with the definition of the DFT and split it into its even and odd components to get

$$\begin{aligned} Y_m &= \sum_{n=0}^{N-1} X_n \,\mathrm{e}\left(-\frac{nm}{N}\right) \\ &= \sum_{n=0}^{N/2-1} X_{2n} \,\mathrm{e}\left(-\frac{2nm}{N}\right) + \sum_{n=0}^{N/2-1} X_{2n+1} \,\mathrm{e}\left(-\frac{(2n+1)m}{N}\right). \end{aligned}$$

We now bring the factor $\mathrm{e}\left(-\frac{m}{N}\right)$ out of the right hand sum to get

$$= \sum_{n=0}^{N/2-1} X_{2n} \,\mathrm{e}\left(-\frac{2nm}{N}\right) + \mathrm{e}\left(-\frac{m}{N}\right) \sum_{n=0}^{N/2-1} X_{2n+1} \,\mathrm{e}\left(-\frac{2nm}{N}\right).$$

Now each of these sums is a length $N/2$ DFT, first on the even numbered elements of X, then on the odds. In the case of the odds, there is also a multiplication by a primitive root of unity. □

Theorem 3.2.4. *If N is a power of 2, then we can compute the DFT of a length N vector of complex values with $\mathcal{O}(N \log N)$ operations and $\mathcal{O}(N)$ space.*

Proof. Write $C(n)$ for the cost of a length n DFT in terms of multiplications and additions. Then by Theorem 3.2.3 we have for even N

$$C(N) = 2C(N/2) + N.$$

Now if N is a power of two, we can continue to split the DFTs until we reach length 2. We have $C(2) = \mathcal{O}(1)$ and so $C(N) = \mathcal{O}(N \log_2 N)$. The space required is $\mathcal{O}(N)$ for the vectors X and Y. □

Practical implementations usually replace the recursion with iteration and perform the transform in place, overwriting the input vector with the result, but this does not effect the overall complexity.

3.2.4 Discrete (Circular) Convolution

The discrete convolution of two length N vectors X and Y is the length N vector $Z = X * Y$ such that for $m \in [0, N-1]$ we have

$$Z_m = \sum_{n=0}^{N-1} X_n Y_{(m-n) \bmod N}$$

We note that padding X and Y with zeros to length $\geq 2N - 1$ will eliminate any overlap and thus compute a linear convolution.

Also, by the discrete version of the Convolution Theorem [15], to compute $X * Y$, we compute the DFTs of X and Y, multiply them term-wise and perform an iDFT on the result. Thus convolution of sequences of length N a power of 2 can be performed in $\mathcal{O}(N \log N)$ time and $\mathcal{O}(N)$ space.

3.2.5 Bluestein's Algorithm

So far, the FFTs described have been limited to vectors of length a power of 2. It is relatively simple to extend these algorithms to handle powers of other

small primes but we will require a FFT algorithm for arbitrary (even large prime) lengths.

One such algorithm was described by Rader [67] but we employ that of Bluestein [9]. Again, we start with X, a length N vector of complex values and aim to compute Y, its DFT via Equation 3.2.1. Now replacing nm with $-\frac{(m-n)^2}{2} + \frac{n^2}{2} + \frac{m^2}{2}$ we get

$$Y_m = \mathrm{e}\left(-\frac{m^2}{2N}\right) \sum_{n=0}^{N-1} X_n \,\mathrm{e}\left(-\frac{n^2}{2N}\right) \mathrm{e}\left(\frac{(m-n)^2}{2N}\right),$$

which is the convolution of $X_n \,\mathrm{e}\left(-\frac{n^2}{2N}\right)$ with $\mathrm{e}\left(\frac{n^2}{2N}\right)$, followed by multiplication by $\mathrm{e}\left(-\frac{m^2}{2N}\right)$. We pad both sequences with zeros to the next power of 2 greater than $2N-1$ and by the Discrete Convolution Theorem we have the required FFT algorithm for arbitrary N. We note that for a given N, we can pre-compute the DFT of $\mathrm{e}\left(\frac{n^2}{2N}\right)$, so each convolution requires only two DFTs (one forward and one backward).

3.2.6 Multi-Dimensional DFTs

It is a trivial matter to extend the single dimension DFT described above to any finite number of dimensions. Given a d dimensional array of complex values, $X_{n_1,n_2\ldots,n_d}$, where each n_i runs from $0 \ldots N_i - 1$, the result of a d dimensional DFT is $Y_{m_1,m_2\ldots,m_d}$ where each m_i also runs from $0 \ldots N_i - 1$, such that

$$Y_{m_1,m_2\ldots,m_d} = \sum_{n_1=0}^{N_1-1} \left(\mathrm{e}\left(\frac{-m_1 n_1}{N_1}\right) \sum_{n_2=0}^{N_2-1} \left(\mathrm{e}\left(\frac{-m_2 n_2}{N_2}\right) \ldots \right) \right).$$

If we set $N = \prod_{i=1}^{d} N_i$ then this is achieved through $\frac{N}{N_1}$ length N_1 DFTs, $\frac{N}{N_2}$ length N_2 DFTs and so on, with a total complexity equivalent to a single length N DFT.

The main issue from an implementation point of view is that all but one of the dimensions will non-contiguous in memory which makes the DFT unlikely to be cache friendly. There is potentially something to be gained, therefore,

from ensuring the contiguous dimension is the largest, but we did not go to these lengths.

3.2.7 Real DFTs

When the input X to a DFT is real valued, the output Y exhibits Hermitian symmetry in that Y_0 and $Y_{N/2}$ are real and for $n \in [1, N/2 - 1]$ we have $Y_n = \overline{Y_{N-n}}$. This can be exploited to yield roughly a 2 fold improvement in time and space.

Theorem 3.2.5. *Starting with two real valued vectors V and W both of length N, form the complex valued length N vector X such that $\Re X_n = V_n$ and $\Im X_n = W_n$ for $n \in [0, N-1]$. Performing a length N DFT on X results in the vector Y from which we recover the n'th element of the DFT of V by*

$$\frac{1}{2}\left[Y_n + \overline{Y_{N-n}}\right]$$

and the n'th element of the DFT of W by

$$\frac{-i}{2}\left[Y_n - \overline{Y_{N-n}}\right]$$

where $n \in [0, N/2]$ in both cases and Y_N is taken to be Y_0. The remaining elements are obtained by conjugation.

Proof. See [80]. □

If our starting point is a single length $2N$ real valued vector, then we split it into odds and evens, apply Theorem 3.2.5 and then Theorem 3.2.3.

This can easily be inverted to yield an algorithm for the iDFT of a vector with Hermitian symmetry with the same (roughly) factor of 2 saving over the naive method.

3.2.8 Computing Large DFTs

The computational efficiency of the FFT algorithms make it possible to consider very large data-sets. However, this eventually starts to cause problems

with the $\mathcal{O}(N)$ space demands of the algorithms. Some respite, at the cost of efficiency, can be obtained by performing the top few levels of the decimation in time algorithm on disk. Specifically, if we wish to compute a length N DFT with N even, we compute in memory the length $N/2$ DFTs of the odd and even elements of the input and store both results to disk. We then combine the results into a single output using Theorem 3.2.3. The combination itself is $\mathcal{O}(N)$ in time (albeit the implied constant is large because of the relative speed of disk versus memory) but $\mathcal{O}(1)$ in space. This method trivially scales to N divisible by larger powers of 2.

3.2.9 Dirichlet Characters and the DFT

Theorem 3.2.6. *For $q \in \mathbb{Z} \geq 3$ and given $\varphi(q)$ complex values $a(n)$ for $n \in [1, q-1]$ and $(n, q) \neq 0$, we can compute*

$$\sum_{n=1}^{q-1} a(n)\chi_{m,q}(n)$$

for the $\varphi(q)$ characters $\chi_{m,q}$ in $\mathcal{O}(\varphi(q)\log(q))$ time and $\mathcal{O}(\varphi(q))$ space.

Proof. Let $U(R)$ be the group of units of the ring R. For $q \in \mathbb{Z}_{>0}$ with the prime decomposition $q = 2^\alpha \prod_{i=1}^{m} p_i^{\alpha_i}$ we consider four cases.

1. $\alpha = 0$ (q is odd) then by the Chinese Remainder Theorem (CRT) we have the constructive, canonical group isomorphism

   $$U(\mathbb{Z}/q\mathbb{Z}) \cong \prod_{i=1}^{m} U(\mathbb{Z}/p_i^{\alpha_i}\mathbb{Z}).$$

   Each of these groups is cyclic so given a primitive root for each $p_i^{\alpha_i}$ we have our construction. Thus this case reduces to performing $\varphi(q)/\varphi(p_i^{\alpha_i})$ length $\varphi(p_i^{\alpha_i})$ DFTs for $i = 1 \ldots m$.

2. $\alpha = 1$ then by the CRT we have the constructive group isomorphism

   $$U(\mathbb{Z}/q\mathbb{Z}) \cong U(\mathbb{Z}/2p_1^{\alpha_1}\mathbb{Z}) \prod_{i=2}^{m} U(\mathbb{Z}/p_i^{\alpha_i}\mathbb{Z}).$$

Each of these groups is cyclic so given a primitive root for $2p_1^{\alpha_1}$ and each $p_i^{\alpha_i}$ $(i > 1)$ we have our construction. Thus this case reduces to performing $\varphi(q)/\varphi(2p_1^{\alpha_1})$ length $\varphi(2p_1^{\alpha_1})$ DFTs followed by $\varphi(q)/\varphi(p_i^{\alpha_i})$ length $\varphi(p_i^{\alpha_i})$ DFTs for $i = 2 \dots m$.

3. $\alpha = 2$ then by the CRT we have the constructive, canonical group isomorphism
$$U(\mathbb{Z}/q\mathbb{Z}) \cong U(\mathbb{Z}/4\mathbb{Z}) \prod_{i=1}^{m} U(\mathbb{Z}/p_i^{\alpha_i}\mathbb{Z}).$$
Each of these groups is cyclic so given a primitive root for each $p_i^{\alpha_i}$ $(i > 1)$ we have our construction. Thus this case reduces to performing $\varphi(q)/2$ length 2 DFTs followed by $\varphi(q)/\varphi(p_i^{\alpha_i})$ length $\varphi(p_i^{\alpha_i})$ DFTs for $i = 1 \dots m$.

4. $\alpha > 2$ then by the CRT we have the constructive, canonical group isomorphism
$$U(\mathbb{Z}/q\mathbb{Z}) \cong U(\mathbb{Z}/2^{\alpha}\mathbb{Z}) \prod_{i=1}^{m} U(\mathbb{Z}/p_i^{\alpha_i}\mathbb{Z}).$$
Now $U(\mathbb{Z}/2^{\alpha}\mathbb{Z})$ is the product of a cyclic group of order 2 and a cyclic group of order $2^{\alpha-2}$ with pseudo primitive roots -1 and 5 respectively. The remaining groups (if there are any) are cyclic so given a primitive root for each $p_i^{\alpha_i}$ $(i > 1)$ we have our construction. Thus this case reduces to performing $\varphi(q)/2$ length 2 DFTs, $\varphi(q)/2^{\alpha-2}$ length $2^{\alpha-2}$ DFTs followed by $\varphi(q)/\varphi(p_i^{\alpha_i})$ length $\varphi(p_i^{\alpha_i})$ DFTs for $i = 1 \dots m$.

In each case, given the ability to perform a length n DFT in time $\mathcal{O}(n \log n)$, we have the claimed overall complexity. □

3.2.10 Factorisation and Finding Primitive Roots

To be able to apply Theorem 3.2.6 for characters of a given modulus q, we require the following.

- The prime factorisation of q.

- If $\alpha = 1$ we need primitive roots for (say) $2p_1^{\alpha_1}$ and each $p_i^{\alpha_i}$ for $\alpha > 1$.
- If $\alpha \neq 1$ we need primitive roots each $p_i^{\alpha_i}$ for $i \geq 1$.

The existence of such roots is guaranteed.

We use the following algorithm to factorise all $q \leq Q$.

- For each $q \in [2, Q]$ set $\text{factor}(q) \leftarrow 0$
- Perform a sieve of Eratosthenes (see section 6.4.1.2), but instead of crossing out multiples of the sieving prime p, test to see if $\text{factor}(np) = 0$, and if so set it to p. At completion, $\text{factor}(q) = 0$ iff q is prime, otherwise it is the smallest prime divisor of q.
- For $q \in [2, Q]$, if $\text{factor}(q) = 0$, then q is prime and we have its factorisation. Otherwise, the factorisation of q is $\text{factor}(q)$ multiplied by the factorisation of $q/\text{factor}(q)$.

To find a primitive root for a prime $p > 4$ we use the following algorithm.

- Factorise $p - 1$. Call its prime factors p_i.
- For $a \in [2, p-1]$, when $(a, p-1) = 1$ compute $a^{\frac{p-1}{p_i}}$ modulo p. If for any p_i this is 1, then a is not a primitive root.

Now to compute the primitive roots (See, for example, [56] ¶ 1.5).

- For each $q \in [5, Q]$ set $\text{pr}(q) \leftarrow 0$.
- Set $\text{pr}(2) \leftarrow 1$, $\text{pr}(3) \leftarrow 2$ and $\text{pr}(4) \leftarrow 3$.
- For $q \in [5, Q]$, if q is composite, skip it. If not
  - Find a primitive root g for q using the algorithm above and set $\text{pr}(q) \leftarrow g$.
  - If g is odd, set $\text{pr}(2q^n) \leftarrow g$ for all $n \geq 1$ and $2q^n \leq Q$, otherwise set $\text{pr}(2q^n) \leftarrow g + p^n$.

- If $g^{q-1} \equiv 1$ modulo q^2, set $g \leftarrow g + q$.
- Set $\mathrm{pr}(q^n) \leftarrow g$ for all $n > 1$ such that $q^n \leq Q$.

Chapter 4

Existing Methods

There now follows a brief summary of methods to compute values of Riemann's Zeta function and Dirichlet L-functions. We start by recalling the standard technique of Euler-Maclaurin summation and applying it to ζ and L_χ.

4.1 Euler-Maclaurin Summation

Theorem 4.1.1. *(Euler-Maclaurin Summation) Let g be a continuous function on $[a,b]$ and $2K+1$ times differentiable there. Let B_n be the n'th Bernoulli number and $B_n(t)$ be the n'th Bernoulli polynomial. Then*

$$\begin{aligned}\sum_{a<n\leq b} g(n) = &\int_a^b g(t)\,\mathrm{d}t + \frac{(g(b)-g(a))}{2} \\ &+ \sum_{k=1}^{K} \frac{B_{2k}}{(2k)!}\left(g^{(2k-1)}(b) - g^{(2k-1)}(a)\right) \\ &- \frac{1}{(2K)!}\int_a^b B_{2K}\left(\{t\}\right) g^{(2K)}(t)\,\mathrm{d}t.\end{aligned}$$

Proof. See, for example, section 2.2.2 of [75]. □

Lemma 4.1.2. *For* $\Re s > 1 - 2K$

$$\begin{aligned}\zeta(s,\alpha) = &\sum_{n=0}^{N} \frac{1}{(n+\alpha)^s} + \frac{(N+\alpha)^{1-s}}{s-1} - \frac{(N+\alpha)^{-s}}{2} \\ &+ \sum_{k=1}^{K} \frac{B_{2k} s(s+1)\dots(s+2k-2)}{(2k)!(N+\alpha)^{s+2k-1}} \\ &+ R_{N,K}.\end{aligned}$$

Furthermore, $R_{N,K}$ *is less in absolute terms than the absolute size of the* $k = K$*'th term of the sum multiplied by*

$$\frac{|s+2K-1|}{\Re s + 2K - 1}.$$

Proof. We start with the series definition for $\zeta(s,\alpha)$ and with $\Re s > 1$. Splitting off the first $N+1$ terms of the sum, we get Equation 4.1.1. However, since the integral remainder converges now for $\Re s > 1 - 2K$, this gives us the analytic continuation of $\zeta(s,\alpha)$ to that enlarged half plane. Further, since for $x \in [0,1]$ and $k \in \mathbb{Z}_{>0}$ we have $B_{2k}(x) \leq B_{2k}$ (see 23.1.13 of [1]) we can bound the absolute size of the remainder as shown. □

Lemma 4.1.3. *For* $\Re s > 1 - 2K$

$$\begin{aligned}\zeta(s) = &\sum_{n=1}^{N} n^{-s} + \frac{N^{1-s}}{s-1} - \frac{N^{-s}}{2} \\ &+ \sum_{k=1}^{K} \frac{B_{2k} s(s+1)\dots(s+2k-2) N^{-s-2k+1}}{(2k)!} \\ &+ R_{N,K}.\end{aligned}$$

Furthermore, $R_{N,K}$ *is less in absolute terms than the absolute size of the* $k = K$*'th term of the sum times*

$$\frac{|s+2K-1|}{\Re s + 2K - 1}.$$

Proof. Identical to Lemma 4.1.2. □

Lemma 4.1.3 gives us the Euler-Maclaurin summation formula applied to ζ. We note that for a given s, we need to take about $|s|$ terms in the initial

sum to obtain any sensible level of precision. Thus, for a single evaluation of $\zeta\left(\frac{1}{2}+it\right)$, this algorithm has time complexity $\mathcal{O}(t)$ and needs space $\mathcal{O}(1)$. It is easy to implement this algorithm rigorously and it is the tool of choice for single evaluations of ζ for $|s|$ not too large.

Lemma 4.1.2 and Equation 2.4.1 together give us a simple method for computing Dirichlet L-functions. To compute a single value for a character of modulus q will require $\mathcal{O}(q|s|)$ operations but by using Theorem 3.2.6 we can compute all the $\varphi(q)$ values of $L_\chi(s)$ using an average of $\mathcal{O}(|s|\log(q))$ operations each. In the former case, the space required is $\mathcal{O}(1)$ and in the latter the improved time complexity comes at the cost of a space requirement that is $\mathcal{O}(\varphi(q))$.

4.2 The Riemann-Siegel Formula for ζ

Theorem 4.2.1. *The Riemann-Siegel formula. Define $\theta(t)$ by*

$$\theta(t) := \frac{1}{2}\left(\Im\left(\log\Gamma\left(\frac{1}{4}+\frac{it}{4}\right)-\log\Gamma\left(\frac{1}{4}-\frac{it}{4}\right)\right)-t\log\pi\right)$$

so that

$$Z(t) := \exp(i\theta(t))\zeta\left(\frac{1}{2}+it\right)$$

is real. Then for $t > 2\pi$, $a = \left(\frac{t}{2\pi}\right)^{1/2}$, $N = \lfloor a \rfloor$ and $\rho = \{a\}$ we have

$$Z(t) = 2\sum_{n=1}^{N} n^{-1/2}\cos(t\log n - \theta(t)) + R(t)$$

where

$$R(t) := \frac{(-1)^{N+1}}{a^{1/2}}\sum_{r=0}^{m}\frac{C_r(\rho)}{a^r} + R_m(t).$$

The C_r are in turn given by

$$C_0(\rho) = \phi(\rho) = \frac{\cos\left(2\pi\left(\rho^2-\rho-\frac{1}{16}\right)\right)}{\cos(2\pi\rho)}$$

$$C_1(\rho) = \frac{-\phi^{(3)}(\rho)}{96\pi^2}$$

$$\ldots$$

and $R_m = \mathcal{O}\left(t^{-\frac{2m+3}{4}}\right)$.

Proof. See, for example [30]. □

Thus computing ζ on the 1/2 line using Rieman-Siegel involves $\mathcal{O}(t^{1/2})$ terms of the main sum followed by correction terms. Gabcke [32] worked out explicit bounds for R_m for $m \leq 10$ making it suitable for rigorous computations. Recently, de Reyna [27] has extended this to general s. Computationally, Riemann-Siegel is a significant advance over Euler-Maclaurin, being $\mathcal{O}(t^{1/2})$ in time whilst remaining $\mathcal{O}(1)$ in space. It was used by Wedeniwski distributed across many machines to verify RH to the 900 000 000 000'th zero [88].

4.3 The Riemann-Siegel Formula for L_χ

An equivalent Riemann-Siegel formula for Dirichlet L-functions was described in [78]. Its main sum has approximately $q\sqrt{t/2\pi q}$ terms so its time complexity is $\mathcal{O}((qt)^{1/2})$. We do not believe that anyone has developed explicit bounds for the remainder terms, which would be essential for rigorous computation.

4.4 The Approximate Functional Equation

The approximate functional equation, and its smoothed version which aims to overcome the catastrophic cancellation caused by the decay of the gamma function away from the real line, is a more general purpose tool applicable to any L-function whose functional equation and Dirichlet coefficients are known (see for example [75]). It requires the computation of a sum involving $\mathcal{O}(t^{1/2})$ terms and $\mathcal{O}(1)$ space. Rubinstein's "lcalc" [74] provides a non-rigorous implementation of this algorithm using IEEE double precision floating point. We note that producing a rigorous implementation, for ζ and Dirichlet L-functions at least, would hinge on being able to compute explicit error bounds for computations involving the incomplete gamma function for complex argument, a

topic which Molin addresses in [55].

4.5 Other Single Value Algorithms for ζ

Another algorithm, with superficial similarities to Rieman-Siegel was described by Berry and Keating [8]. However, a number of algorithms for computing single values of ζ have been developed with better asymptotic time complexity than $\mathcal{O}(t^{1/2})$.

In particular, Hiary in [38] cites a method due to Heath-Brown where the exponent is 1/3 and then goes on to describe three other algorithms with exponents 2/5, 1/3 and 4/13. The last of these relies on a fast way of computing cubic exponential sums using the FFT.

4.6 Other Single Value Algorithms for L_χ

In addition to computing Dirichlet L-functions as outlined above, various techniques have been devised to compute single values. Rumely [76] used a method based on Euler-Maclaurin but employing polynomial approximations to the Taylor expansions

$$L_\chi(s) = \sum_{n=0}^{\infty} a_n(s_0, \chi)(s - s_0)^n$$

with s_0 on the critical line.

Coffey [23] describes an efficient algorithm for Hurwitz Zeta which is "particularly useful if the domain of interest does not lie far from the real axis" which makes it of little interest for our purposes. Similarly, techniques described by Slezeviien [79] (complexity somewhere between Euler-Maclaurin and Riemann-Siegel) have no particular advantages for our application.

4.7 The Odlyzko-Schönhage Algorithm

This algorithm [62] is designed to be efficient when computing many values of ζ simultaneously. This efficiency is achieved by sharing parts of the computation across many s which is sometimes referred to as "recycling" [14]. This is achieved by expressing the Riemann-Siegel formula for many equally spaced points on the critical line as a DFT and then exploiting the efficiency of the FFT algorithm. It has time complexity of $\mathcal{O}(t^\epsilon)$ on average per value and requires $\mathcal{O}(t^{1/2})$ space when working at height t. This algorithm has been used for the large scale computations of ζ by Odlyzko [61] and Gourdon. [36].

4.8 Booker's Algorithm

In [13] section 5, Booker describes a rigorous DFT based algorithm for computing L-functions at many equally spaced points on the critical line simultaneously. Starting with the Fourier transform of the L-function, Booker shifts the line of integration right (collecting residues at any poles on the way) until the L-function can be expressed in terms of its defining series. Computing an approximation to this sum, and to its inverse Fourier transform to recover the L-function, are both achieved efficiently by recourse to the FFT.

The overall time complexity is on average $\mathcal{O}(t^\epsilon)$ (the same as Odlyzko-Schönhage) and it requires space of $\mathcal{O}(t)$ (rather than $\mathcal{O}(t^{1/2})$). It has been used to compute values of Artin L-functions (by Booker) and of Dirichlet L-functions (see section 7.1).

Chapter 5

Windowing ζ

5.1 Overview

We now describe a new algorithm for calculating Riemann's zeta function on the critical line. It matches the asymptotic time complexity of those due to Odlyzko-Schönhage (see section 4.7) and Booker (see section 4.8) while simultaneously overcoming the difficulties of producing a rigorous implementation of the former without incurring the larger space demands of the latter. This is achieved essentially by windowing ζ with a Gaussian centred high up the critical line and then applying the ideas behind Booker's algorithm.

We aim to compute

$$f(t) := \pi^{-\frac{i(t+t_0)}{2}} \Gamma\left(\frac{\frac{1}{2} + i(t+t_0)}{2}\right) \exp\left(\frac{\pi(t+t_0)}{4} - \frac{t^2}{2h^2}\right) \zeta\left(\frac{1}{2} + i(t+t_0)\right) \tag{5.1.1}$$

with $t \in \mathbb{R}$ and $t_0, h > 0$.

We proceed as follows:

1. Select $t_0, h > 0$, $K \in \mathbb{Z}_{\geq 0}$ and $A, B > 0$ such that $N = AB \in 2\mathbb{Z}_{>0}$. For $n = -\frac{N}{2} \ldots \frac{N}{2} - 1$ and $k = 0, 1, \ldots, K$ compute $g\left(\frac{n}{A}; k\right)$ where $g(t; k)$ is defined by

$$g(t; k) := \Gamma\left(\frac{\frac{1}{2} + i(t+t_0)}{2}\right) \exp\left(\frac{\pi(t+t_0)}{4} - \frac{t^2}{2h^2}\right) (-2\pi i t)^k.$$

2. By adding an appropriate error term, approximate

$$\widetilde{g}(n;k) := \sum_{l\in\mathbb{Z}} g\left(\frac{n}{A} + lB; k\right).$$

3. Use discrete Fourier transforms to compute

$$\widetilde{G}^{(k)}(m) := \sum_{l\in\mathbb{Z}} G^{(k)}\left(\frac{m}{B} + lA\right),$$

where

$$G(u) := \int_{-\infty}^{\infty} g(t;0)\,\mathrm{e}(-tu)\,\mathrm{d}t.$$

4. Add an appropriate error term to recover $G^{(k)}\left(\frac{m}{B}\right)$ from $\widetilde{G}^{(k)}(m)$.

5. Use a series of convolutions to sum terms involving $G\left(\frac{m}{B}\right)$ and its derivatives yielding an approximation to $F\left(\frac{m}{B}\right)$, where

$$F(x) := \int_{-\infty}^{\infty} f(t)\,\mathrm{e}(-tx)\,\mathrm{d}t.$$

6. By adding an appropriate error term, approximate

$$\widetilde{F}(m) := \sum_{l\in\mathbb{Z}} F\left(\frac{m}{B} + lA\right).$$

7. Now use another discrete Fourier transform to compute

$$\widetilde{f}(n) := \sum_{l\in\mathbb{Z}} f\left(\frac{n}{A} + lB\right).$$

8. Finally, add another error term to recover $f\left(\frac{n}{A}\right)$ from $\widetilde{f}(n)$.

5.2 Computing $g\left(\frac{n}{A};k\right)$

The only intricacy in computing g is that of computing $\Gamma\left(\frac{1}{4}+ix\right)$ for real x. We use the following lemma.

Lemma 5.2.1. *For $N \in \mathbb{Z}_{>0}$ and $x \in \mathbb{R}$, write $z = \frac{1}{4} + ix$. Then we have*

$$\log \Gamma(z) = \left(z - \frac{1}{2}\right) \log z - z + \frac{1}{2} \log(2\pi) + \sum_{n=1}^{N} \frac{B_{2n}}{2n(2n-1)z^{2n-1}} + R_N.$$

Furthermore, the absolute value of R_N is less than the absolute value of the $n = N$'th term of the sum, multiplied by 2^N.

Proof. We use Olver's bound (Equation 4.1 of [37]) for the error in truncating Stirling's approximation to $\log \Gamma$. □

5.3 Approximating $\widetilde{g}$ with g

We compute values of $g\left(\frac{n}{A}; k\right)$ for a given k which we intend to use as approximations to $\widetilde{g}\,(n; k)$. The following lemmas bound the error introduced.

Lemma 5.3.1. *Define the incomplete Gamma function for $\Re s > 0$ by [1]*

$$\Gamma(s, x) := \int_x^\infty t^{s-1}\, \mathrm{e}^{-t}\, \mathrm{d}t.$$

Then, given $\kappa > -1$ and $x, h > 0$, we have

$$\int_x^\infty w^\kappa \exp\left(\frac{-w^2}{2h^2}\right) \mathrm{d}w = 2^{\frac{\kappa-1}{2}} h^{\kappa+1} \Gamma\left(\frac{\kappa+1}{2}, \frac{x^2}{2h^2}\right).$$

Proof. Substitute $t = \frac{w^2}{2h^2}$. □

Lemma 5.3.2. *For $k \in \mathbb{Z}_{\geq 0}$, $t \in \mathbb{R}$ and $t_0, h > 0$, recall that we define g by*

$$g(t; k) := \Gamma\left(\frac{\frac{1}{2} + i(t + t_0)}{2}\right) \exp\left(\frac{\pi(t + t_0)}{4} - \frac{t^2}{2h^2}\right) (-2\pi i t)^k.$$

Then

$$|g(t; k)| \leq 4|2\pi t|^k \exp\left(\frac{-t^2}{2h^2}\right).$$

Proof. We use the trivial bound $\left|\Gamma\left(\frac{\frac{1}{2}+ix}{2}\right) \mathrm{e}^{\frac{\pi x}{4}}\right| < 4$. □

Lemma 5.3.3. *Let $n \in \left[-\frac{N}{2}, \frac{N}{2} - 1\right]$ and $B > h\sqrt{k}$. Then*

$$\left| \sum_{l \in \mathbb{Z}_{\neq 0}} g\left(\frac{n}{A} + lB; k\right) \right| \leq$$
$$8(\pi B)^k \left[\exp\left(\frac{-B^2}{8h^2}\right) + 2^{\frac{3k-1}{2}} \left(\frac{h}{B}\right)^{k+1} \Gamma\left(\frac{k+1}{2}, \frac{B^2}{8h^2}\right) \right].$$

Proof. We consider the right tail from $n = -\frac{N}{2}$. The first term missing is $g\left(\frac{B}{2}; k\right)$ and $\frac{B}{2}$ is sufficiently large that our bound for $g(t; k)$ (Lemma 5.3.2) is decreasing. Thus we can split off the first term and majorise the balance with an integral. This process results in

$$2 \left[\left| g\left(\frac{B}{2}; k\right) \right| + \int_1^\infty 4(\pi B(2w-1))^k \exp\left(\frac{-(2w-1)^2 B^2}{8h^2}\right) \mathrm{d}w \right].$$

We now apply Lemma 5.3.1 to the integral and the result follows. □

Thus, appealing to Lemma 5.3.3 and chosing the parameters B and h, we can control the error introduced by using $g\left(\frac{n}{A}; k\right)$ in place of $\widetilde{g}\left(n; k\right)$.

5.4 Computing $\widetilde{G}$ from $\widetilde{g}$

We now wish to compute values of $\widetilde{G}^{(k)}(m)$ from $\widetilde{g}(n; k)$. The following lemma provides an efficient mechanism to achieve this.

Lemma 5.4.1. *Up to a constant factor, the functions $\widetilde{g}(n; k)$ and $\widetilde{G}^{(k)}(m)$ form a discrete Fourier transform pair of length N.*

Proof. It is a standard result that given f with Fourier transform F, the Fourier transform of $x^k f(x)$ is $\left(\frac{i}{2\pi}\right)^k \frac{\mathrm{d}^k F(u)}{\mathrm{d}u^k}$. Thus $G^{(k)}(u)$ is the Fourier transform of $g(t; k)$ and the result follows from Theorem 3.2.2. □

5.5 Approximating $G^{(k)}$ with $\widetilde{G}^{(k)}$

The result of the DFT above is N values of $\widetilde{G}^{(k)}(m)$. The following lemmas bound the error in using these values in place of $G\left(\frac{m}{B}\right)$.

Lemma 5.5.1. *For $\sigma \in 2\mathbb{Z}_{>0}+1$ define*

$$C(\sigma,t_0,h,k) :=$$
$$(2\pi)^k \int_{-\infty}^{\infty} \left|\Gamma\left(\frac{\sigma+i(t+t_0)}{2}\right)\exp\left(\frac{\pi(t+t_0)}{4}-\frac{t^2}{2h^2}\right)\left(\frac{1}{2}+\sigma-it\right)^k\right| \mathrm{d}t.$$

Then for $t_0 > \sigma+\frac{1}{2}$ writing

$$X := h^{m+1}2^{\frac{\sigma-3-4k}{4}}(2\sigma+1)^k\left(\Gamma\left(\frac{\sigma+1}{4}\right)-\Gamma\left(\frac{\sigma+1}{4},\frac{(2\sigma+1)^2}{8h^2}\right)\right)$$

and

$$Y := h^{\frac{\sigma+1+2k}{2}}2^{\frac{\sigma-3+2k}{4}}\Gamma\left(\frac{\sigma+1+2k}{4},\frac{(2\sigma+1)^2}{8h^2}\right)$$

we have

$$C(\sigma,t_0,h,k) \le 2^{\frac{6k+7-\sigma}{4}}\pi^{\frac{2k+1}{2}}\,\mathrm{e}^{\frac{1}{2\sigma}}\sum_{l=0}^{\frac{\sigma-1}{2}}\binom{\frac{\sigma-1}{2}}{l}t_0^{\frac{\sigma-2l-1}{2}}\left[X+Y\right].$$

Proof.

$$(2\pi)^k \int_{-\infty}^{\infty} \left|\Gamma\left(\frac{\sigma+i(t+t_0)}{2}\right)\exp\left(\frac{\pi(t+t_0)}{4}-\frac{t^2}{2h^2}\right)\left(\frac{1}{2}+\sigma-it\right)^k\right| \mathrm{d}t$$
$$\le 2^{k+1}\pi^k \int_{0}^{\infty} \left|\Gamma\left(\frac{\sigma+i(t+t_0)}{2}\right)\exp\left(\frac{\pi(t+t_0)}{4}-\frac{t^2}{2h^2}\right)\left(\frac{1}{2}+\sigma-it\right)^k\right| \mathrm{d}t$$

We now split the integral so that the upper bound becomes

$$2^{\frac{6k+7-\sigma}{4}}\pi^{\frac{2k+1}{2}}\,\mathrm{e}^{\frac{1}{2\sigma}}\left[\int_0^{\sigma+\frac{1}{2}}(t+t_0)^{\frac{\sigma-1}{2}}(\sigma+1/2)^k\exp\left(\frac{-t^2}{2h^2}\right)\mathrm{d}t + \int_{\sigma+\frac{1}{2}}^{\infty}(t+t_0)^{\frac{\sigma-1}{2}}t^k\exp\left(\frac{-t^2}{2h^2}\right)\mathrm{d}t\right].$$

Now the first integral evaluates to

$$\sum_{l=0}^{\frac{\sigma-1}{2}}\binom{\frac{\sigma-1}{2}}{l}t_0^{\frac{\sigma-2l-1}{2}}h^{l+1}2^{\frac{l-1}{2}}(\sigma+1/2)^k\left(\Gamma\left(\frac{l+1}{2}\right)-\Gamma\left(\frac{l+1}{2},\frac{(2\sigma+1)^2}{8h^2}\right)\right)$$

and the second to

$$\sum_{l=0}^{\frac{\sigma-1}{2}} \binom{\frac{\sigma-1}{2}}{l} t_0^{\frac{\sigma-2l-1}{2}} h^{l+k+1} 2^{\frac{l+k-1}{2}} \Gamma\left(\frac{l+k+1}{2}, \frac{(2\sigma+1)^2}{8h^2}\right).$$

□

Lemma 5.5.2. *Let $\sigma \in 2\mathbb{Z}_{>0}+1$. Then $G^{(k)}(u)$ is bounded in absolute terms by*

$$C(\sigma, t_0, h, k) \exp\left(\frac{(2\sigma+1)^2}{8h^2} - (2\sigma-1)\pi|u|\right) +$$
$$2^{k+2}\pi^{k+1} \exp\left(\frac{-t_0^2}{2h^2}\right) \sum_{l=0}^{\frac{\sigma-1}{2}} \frac{((2l+1/2)^2+t_0^2)^{\frac{k}{2}}}{l!} \exp\left(\frac{(4l+1)^2}{8h^2} - (4l+1)\pi|u|\right).$$

Proof. First we consider $u \geq 0$. We write

$$\left|G^{(k)}(u)\right| = \left| \int_{-\infty}^{\infty} \Gamma\left(\frac{\frac{1}{2}+i(t+t_0)}{2}\right) \exp\left(\frac{\pi(t+t_0)}{4} - \frac{t^2}{2h^2}\right) (-2\pi i t)^k \,\mathrm{e}(-tu)\,\mathrm{d}t \right|.$$

Substituting $s = \frac{1}{2} + i(t+t_0)$, we now move the line of integration right to $\Re(s) = \sigma$ giving

$$\left|G^{(k)}(u)\right| \leq$$
$$\exp\left(\frac{(2\sigma-1)^2}{8h^2} - \pi u(2\sigma-1)\right) \times$$
$$(2\pi)^k \int_{-\infty}^{\infty} \left|\Gamma\left(\frac{\sigma+i(t+t_0)}{2}\right) \exp\left(\frac{\pi(t+t_0)}{4}\right) \exp\left(\frac{-t^2}{2h^2}\right) \left(\frac{1}{2}-\sigma-it\right)^k\right| \mathrm{d}t. \tag{5.5.1}$$

For $u < 0$, we move the line of integration left to $\Re(s) = -\sigma$, picking up the poles of $\Gamma\left(\frac{s}{2}\right)$ at $s = 0, -2, \ldots, 1-\sigma$. These give a contribution bounded by

$$2^{k+2}\pi^{k+1} \exp\left(\frac{-t_0^2}{2h^2}\right) \sum_{l=0}^{\frac{\sigma-1}{2}} \frac{((2l+1/2)^2+t_0^2)^{\frac{k}{2}}}{l!} \exp\left(\frac{(4l+1)^2}{8h^2} + (4l+1)\pi u\right).$$

The integral which remains is now

$$(2\pi)^k \exp\left(\frac{(2\sigma+1)^2}{8h^2} + (2\sigma+1)\pi u\right) \times$$
$$\int_{-\infty}^{\infty} \left|\Gamma\left(\frac{-\sigma+i(t+t_0)}{2}\right) \exp\left(\frac{\pi(t+t_0)}{4} - \frac{t^2}{2h^2}\right) (\sigma+\frac{1}{2}-it)^k\right| \mathrm{d}t.$$

Finally, for our range of σ and for $t \in \mathbb{R}$, we have $|\Gamma(-\sigma/2+it)| < |\Gamma(\sigma/2+it)|$ and the result follows. □

We are free to chose a value of σ that minimises this bound for a particular choice of u. We note that for t_0 large compared to h, $C(\sigma, t_0, h, k)$ is of order $t_0^{\frac{\sigma-1+2k}{2}}$.

Lemma 5.5.3. *For $m \in [0, N/2]$ and $\sigma \in 2\mathbb{Z}_{>0}+1$*

$$\left|\sum_{l\in\mathbb{Z}_{\neq 0}} G^{(k)}\left(\frac{m}{B}+lA\right)\right| \leq 2^{k+3}\pi^{k+1}\exp\left(\frac{-t_0^2}{2h^2}\right)S+$$
$$2\left(1+\frac{1}{A\pi(2\sigma-1)}\right)C(\sigma,t_0,h,k)\exp\left(\frac{(2\sigma+1)^2}{8h^2}-\frac{A\pi(2\sigma-1)}{2}\right)$$

where S is the sum

$$\sum_{l=0}^{\frac{\sigma-1}{2}}\left(1+\frac{1}{A\pi(4l+1)}\right)\frac{((2l+1/2)^2+t_0^2)^{\frac{k}{2}}}{l!}\exp\left(\frac{(4l+1)^2}{8h^2}-\frac{A\pi(4l+1)}{2}\right).$$

Proof. The left tail from $m = \frac{N}{2}$ majorises every case. The first term missing is $G^{(k)}\left(\frac{-A}{2}\right)$ and the remainder of the left tail is less in absolute terms than

$$\int_1^\infty\Bigg[C(\sigma,t_0,h,k)\exp\left(\frac{(2\sigma+1)^2}{8h^2}-\frac{A\pi(2n-1)(2\sigma-1)}{2}\right)+$$
$$2^{k+2}\pi^{k+1}\exp\left(\frac{-t_0^2}{2h^2}\right)\times$$
$$\sum_{l=0}^{\frac{\sigma-1}{2}}\frac{((2l+1/2)^2+t_0^2)^{\frac{k}{2}}}{l!}\exp\left(\frac{(4l+1)^2}{8h^2}-\frac{A\pi(2n-1)(4l+1)}{2}\right)\Bigg]\mathrm{d}n.$$

□

5.6 Computing F from $G^{(k)}$

Now armed with values of $G^{(k)}\left(\frac{m}{B}\right)$ for several k, we wish to compute $F\left(\frac{m}{B}\right)$. We use the following result.

Lemma 5.6.1. *Let F be the Fourier transform of f (Equation 5.1.1). Then*

$$\left| F(x) \quad - \quad \sum_{j=1}^{\infty} \frac{1}{\sqrt{j}} \left(j\sqrt{\pi}\right)^{-it_0} G\left(x + \frac{\log\left(j\sqrt{\pi}\right)}{2\pi}\right) \right| = 2\pi^{\frac{5}{4}} \exp\left(\frac{1}{8h^2} - \frac{t_0^2}{2h^2} - \pi x\right).$$

Proof. We start with $F(x) = \int_{-\infty}^{\infty} f(t)\,\mathrm{e}(-tx)\,\mathrm{d}t$ and substitute $s = \frac{1}{2} + i(t+t_0)$. We then shift the line of integration right to $\Re(s) = \sigma > 1$ picking up the simple pole of $\zeta(s)$ with residue 1 at $s = 1$. Now write $\zeta(s)$ as a sum (over j), interchange the sum and integral and move the line of integration back to $\frac{1}{2}$. □

The following lemma allows us to truncate the sum over j at J.

Lemma 5.6.2. *Let $x \geq 0$. Then*

$$\left| \sum_{j>J} \frac{1}{\sqrt{j}} \left(j\sqrt{\pi}\right)^{-it_0} G\left(x + \frac{\log\left(j\sqrt{\pi}\right)}{2\pi}\right) \right| \leq C(\sigma, t_0, h, 0) \exp\left(\frac{(2\sigma-1)^2}{8h^2}\right) \pi^{\frac{1-2\sigma}{4}} \frac{J^{1-\sigma}}{\sigma - 1}.$$

Proof. Take $x = 0$ and apply Equation 5.5.1 of Lemma 5.5.2. □

This suggests that J will need to be in the region of $(t_0)^{\frac{1}{2}}$.

We need to calculate $F(x)$ on a lattice of points u_m each $\frac{1}{B}$ apart. Using Taylor's Theorem with K terms (see Lemma 5.6.3 below for the truncation error) we can write

$$F(x) \approx \sum_{k=0}^{K-1} \sum_{m} \frac{G^{(k)}(x + u_m)}{k!} S_m^{(k)} \tag{5.6.1}$$

where we set $\xi = \frac{1}{2B}$ and then

$$S_m^{(k)} := \sum_{\frac{\log(j\sqrt{\pi})}{2\pi} \in [u_m - \xi, u_m + \xi)} \frac{1}{\sqrt{j}} \left(j\sqrt{\pi}\right)^{-it_0} \left(\frac{\log\left(j\sqrt{\pi}\right)}{2\pi} - u_m\right)^k.$$

Now for each k, Equation 5.6.1 is a discrete convolution so computing our approximation to $F\left(\frac{m}{B}\right)$ is achieved by summing the output of K such convolutions. The following lemma provides the bound on the error from truncating the Taylor series to K terms.

Lemma 5.6.3. *Let $w \in [-\xi, \xi]$. Then we have*

$$\left|\sum_{k=K}^{\infty} \frac{G^{(k)}(u)w^k}{k!}\right| \leq \frac{2^{\frac{K+5}{2}}\pi^{K+\frac{1}{2}}h^{K+1}\xi^K}{\Gamma\left(\frac{K+2}{2}\right)}.$$

Proof.

$$\begin{aligned}\left|\sum_{k=K}^{\infty} \frac{G^{(k)}(u)w^k}{k!}\right| &\leq \sup_{u'\in(u-\xi,u+\xi]}\left|\frac{G^{(K)}(u')\xi^K}{K!}\right| \\ &\leq \sup_{u'\in(u-\xi,u+\xi]}\left|\int_{-\infty}^{\infty} \frac{g(t;k)\xi^K\,\mathrm{e}(-u't)}{K!}\,\mathrm{d}t\right| \\ &\leq 8\int_0^{\infty} \frac{(2\pi t\xi)^K}{K!}\exp\left(\frac{-t^2}{2h^2}\right)\,\mathrm{d}t \\ &= \frac{2^{\frac{3K+5}{2}}\pi^K\xi^K h^{K+1}\Gamma\left(\frac{K+1}{2}\right)}{\Gamma(K+1)}\end{aligned}$$

and the result follows from the duplication formula for Γ. □

Since this error term occurs J times in Equation 5.6.1, weighted by $\frac{1}{\sqrt{j}}$ each time, we multiply it by $2\sqrt{J}-1$.

5.7 Approximating $\widetilde{F}$ with F

Lemma 5.7.1. *Let $\sigma \in 2\mathbb{Z}+1$ and $1 < \sigma < t_0$. Then we have*

$$\begin{aligned}|F(x)| \leq &\zeta(\sigma)\pi^{\frac{1-2\sigma}{4}}C(\sigma,t_0,h,0)\exp\left(\frac{(2\sigma-1)^2}{8h^2} - \pi|x|(2\sigma-1)\right) \\ &+ 2\pi^{\frac{5}{4}}\exp\left(\frac{1}{8h^2} - \pi|x| - \frac{t_0^2}{2h^2}\right).\end{aligned}$$

Proof. Since $f(t)$ is real, its Fourier Transform $F(x)$ has the property $F(-x) = \overline{F(x)}$ so we need only consider $x \geq 0$. We write $s = \frac{1}{2} + i(t+t_0)$ and shift the line of integration right to $\Re(s) = \sigma$ encountering the pole of $\zeta(s)$ at $s = 1$. This yields a residue smaller in absolute terms than

$$2\pi^{\frac{5}{4}}\exp\left(\frac{1}{8h^2} - \pi x - \frac{t_0^2}{2h^2}\right).$$

The remaining integral is then bounded in exactly the same fashion as in Lemma 5.5.2 using $|\zeta(\sigma+it)| \leq |\zeta(\sigma)|$ for $\sigma > 1$ and $t \in \mathbb{R}$. □

Lemma 5.7.2. *For $n \in [0, \frac{N}{2}]$ we have*

$$\left| \sum_{l \in \mathbb{Z}_{\neq 0}} F\left(\frac{n}{B} + lA\right) \right| \leq$$
$$2\zeta(\sigma)\pi^{\frac{1-2\sigma}{4}} C(\sigma, t_0, h, 0) \exp\left(\frac{(2\sigma-1)^2}{8h^2} - \frac{A\pi(2\sigma-1)}{2}\right)\left(1 + \frac{1}{A\pi(2\sigma-1)}\right)$$
$$+ 4\pi^{\frac{5}{4}} \exp\left(\frac{1-4t_0^2}{8h^2} - \frac{\pi A}{2}\right)\left(1 + \frac{1}{A\pi}\right).$$

Proof. The left tail from $n = N/2$ majorises all other cases. The first term missing is $F\left(\frac{-A}{2}\right)$ and the remaining terms are bounded by

$$\int_1^\infty \left[\zeta(\sigma)\pi^{\frac{1-2\sigma}{4}} C(\sigma, t_0, h, 0) \exp\left(\frac{-\pi A(2\sigma-1)(2n-1)}{2}\right)\right.$$
$$\left. + 2\pi^{\frac{5}{4}} \exp\left(\frac{1-4t_0^2}{8h^2} - \frac{\pi(2n-1)A}{2}\right)\right] \mathrm{d}n.$$

□

5.8 Computing $\widetilde{f}$ from $\widetilde{F}$

We now need to move from $\widetilde{F}$ to $\widetilde{f}$. The following lemma provides the means:

Lemma 5.8.1. *Up to a constant factor, the functions $\widetilde{f}$ and $\widetilde{F}$ form a discrete Fourier transform pair of length N.*

Proof. We defined f and F to be a Fourier transform pair and Theorem 3.2.2 therefore applies. □

Hence we can compute N values of $\widetilde{f}(n)$ from our N approximations to $\widetilde{F}(m)$ efficiently via a single DFT.

5.9 Approximating f with $\widetilde{f}$

The final step is to extract approximations to $f\left(\frac{n}{A}\right)$ from our values of $\widetilde{f}(n)$. The following two lemmas bound the error introduced if we simply equate them.

Lemma 5.9.1. *Given $t \geq 0$ and $t_0 > \exp(\mathrm{e})$, set $\beta = \frac{1}{6} + \frac{\log\log t_0}{\log t_0}$. Then*

$$|f(t)| \leq 12(t+t_0)^{\beta} \exp\left(-\frac{t^2}{2h^2}\right).$$

Proof. By [19] we have for $t + t_0 > \mathrm{e}$

$$\left|\zeta\left(\frac{1}{2} + i(t+t_0)\right)\right| \leq 3(t+t_0)^{\frac{1}{6}} \log(t+t_0) \tag{5.9.1}$$

so with $(t+t_0) > \exp(\mathrm{e})$ and β defined as above

$$(t+t_0)^{\frac{1}{6}} \log(t+t_0) \leq (t+t_0)^{\beta}.$$

The factor of 4 comes from the trivial bound for the Gamma factor. □

Lemma 5.9.2. *For $t \geq 0$ and $t_0 > \exp(\mathrm{e})$ set $\beta = \frac{1}{6} + \frac{\log\log t_0}{\log t_0}$. Then providing $\frac{\beta h^2}{t_0} \leq \frac{B}{2} \leq t_0$ and $n \in [0, N-1]$ we have*

$$\left|\sum_{l \in \mathbb{Z}_{\neq 0}} f\left(\frac{n - N/2}{A} + lB\right)\right| \leq 24(X + \frac{2^{\beta} h}{B}(Y+Z)),$$

where

$$X = \left(\frac{B}{2} + t_0\right)^{\beta} \exp\left(-\frac{B^2}{8h^2}\right),$$

$$Y = (t_0)^{\beta} \sqrt{\frac{\pi}{2}} \left(\operatorname{erf}\left(\frac{t_0}{h\sqrt{2}}\right) - \operatorname{erf}\left(\frac{B}{2h\sqrt{2}}\right)\right)$$

and

$$Z = 2^{\frac{\beta-1}{2}} h^{\beta} \Gamma\left(\frac{\beta+1}{2}, \frac{B^2}{8h^2}\right).$$

Proof. The lower bound on B ensures that the bound of Lemma 5.9.1 is decreasing for $t \geq \frac{B}{2}$. The worst case is when $n = 0$ and for any n, the right tail majorises the left. The first missing term to the right is $f\left(\frac{B}{2}\right)$ and the remaining terms are majorised by

$$\left|12 \int_0^{\infty} \left(\frac{(2w+1)B}{2} + t_0\right)^{\beta} \exp\left(-\frac{\left(\frac{(2w+1)B}{2}\right)^2}{2h^2}\right) \mathrm{d}w\right| \leq$$

$$\frac{12}{B}\left|\int_{\frac{B}{2}}^{t_0} (2t_0)^{\beta} \exp\left(\frac{-t^2}{2h^2}\right) \mathrm{d}t + \int_{t_0}^{\infty} (2t)^{\beta} \exp\left(\frac{-t^2}{2h^2}\right) \mathrm{d}t\right|.$$

The result follows from Lemma 5.3.1. □

5.10 Up-sampling

Having calculated many values of $f(t)$ on a spacing of $\frac{1}{A}$, we compute $\Lambda(t)$ (Equation 2.3.1) by removing the Gaussian factor. We then apply the results of section 2.6 to rigorously up-sample.

We define the function $W : \mathbb{R} \to \mathbb{R}$

$$W(t) := \Lambda(t) \exp\left(\frac{-(t-t_0)^2}{2H^2}\right).$$

Note that the Gaussian width $H > 0$ need not be and indeed will not be the same as the h of Equation 5.1.1.

We aim to estimate $W(t_0)$ from our samples using theorems 2.6.1 (Whittaker-Shannon) and 2.6.2. The following lemmas provide the necessary rigorous bounds.

Lemma 5.10.1. *Define I by*

$$I := 4 \int_{\frac{A}{2}}^{\infty} \left| \int_{-\infty}^{\infty} W(t)\, \mathrm{e}(-xt)\, \mathrm{d}t \right| \mathrm{d}x.$$

Then we have

$$\begin{aligned} I \leq & \frac{4\zeta(\sigma)}{2\sigma-1} \pi^{\frac{-3-2\sigma}{4}} C(\sigma, t_0, H, 0) \exp\left(\frac{(2\sigma-1)^2}{8H^2} - \frac{\pi A(2\sigma-1)}{2}\right) \\ & + 8\pi^{\frac{1}{4}} \exp\left(\frac{1-4t_0^2}{8H^2} - \frac{\pi A}{2}\right). \end{aligned}$$

Proof. The inner integral is $F(x)$ with H in place of h. □

Lemma 5.10.2. *For $t \geq \mathrm{e}$*

$$|W(t)| \leq 12 t^{\frac{1}{6}} \log t \exp\left(\frac{-(t-t_0)^2}{2H^2}\right).$$

Proof. This follows from Equation 5.9.1 of Lemma 5.9.1. □

Lemma 5.10.3. *For $t \geq 0$ and $t_0 > \exp(\mathrm{e})$, set $\beta = \frac{1}{6} + \frac{\log\log t_0}{\log t_0}$. Then if $\frac{\beta H^2}{t_0} \leq \frac{N_s}{A} \leq t_0$ we have*

$$\left| \sum_{\left|n - \frac{t_0}{A}\right| > N_s} W\left(\frac{n}{A}\right) \operatorname{sinc}\left(A\pi\left(\frac{n}{A} - t_0\right)\right) \right| \leq \frac{24}{\pi}(X + 2^{\beta-1}(Y+Z)),$$

where

$$X = \frac{A}{N_s}\left(\frac{N_s}{A} + t_0\right)^{\beta} \exp\left(-\frac{N_s^2}{2A^2h^2}\right),$$
$$Y = (t_0)^{\beta}\left[\Gamma\left(0, \frac{N_s^2}{2A^2H^2}\right) - \Gamma\left(0, \frac{t_0^2}{2H^2}\right)\right],$$

and

$$Z = 2^{\frac{\beta}{2}} h^{\beta} \Gamma\left(\frac{\beta}{2}, \frac{t_0^2}{H^2}\right).$$

Proof. The proof follows almost identical lines to that of Lemma 5.9.2 □

5.11 Choice of Parameters

We implemented this windowed zeta algorithm in the 'C' programming language using the GNU C Compiler (GCC [81]) under Unix. Multiple precision interval arithmetic was used throughout and we relied on the MPFI package [68].

For computational reasons, we found it expedient to introduce an extra up-sampling step by a factor of 32, achieved through padding the $F(x)$ vector with zeros (actually a very small error estimated using Lemma 5.7.1) before the final DFT.

The various parameters used were calculated experimentally, based on a desire to achieve 101 binary digits of absolute accuracy in the location of the zeros (eventually) to a height of 3×10^{10} up the critical line and with a sufficiently dense spacing to resolve all the zeros (possibly after further up-sampling). The value of N was the largest power of 2 that allowed the process to run in < 1 Gbyte of memory as this allowed us to make best use of the multi-core CPUs we had available to us.

- $PREC = 300$ the working precision in bits
- $t_0 = 3 \times 10^{10}$ the maximum height up the critical line
- $t_{min} = 5\,000$ the minimum height up the critical line

- $N1 = 32\,768 = 2^{15}$ the number of data points before up-sampling
- $N = 1\,048\,576 = 2^{20}$ the number of data points after up-sampling
- $B = 5\,376$ the width of the window
- $h = \frac{176\,431}{2\,048}$ the Gaussian width
- $J = 103\,000$ the number of terms to sum when computing $F(x)$
- $K = 44$ the number of differentials of G to use when computing $F(x)$

We note that the values chosen for N and B imply an output spacing of $\frac{21}{4\,096}$ which is about 58 times as dense as the anticipated zero spacing. Experience suggests that with rigorous up-sampling, a density of $5 - 10$ times the zero spacing would suffice but the extra density was achieved effectively for free. Also, only the central 2 100 of each window of width $B = 5\,376$ is actually usable due to the decay of the Gaussian outside that.

Using the analysis above, we can now bound the necessary error terms.

- The error approximating $\tilde{g}$ with g (Lemma 5.3.3) is $< 3.2 \times 10^{-82}$.
- The error approximating $\tilde{G}$ with G (Lemma 5.5.3) is $< 2.3 \times 10^{-213}$.
- The error approximating F by the truncated sum (Lemmas 5.6.1 and 5.6.2) total $< 1.7 \times 10^{-83}$.
- The error truncating the Taylor approximation to G to $K = 44$ terms (Lemma 5.6.3) is $< 1.5^{-82}$
- The error approximating $\tilde{F}$ with F (Lemma 5.7.2) is $< 10^{-307}$ (actually much smaller but we want to avoid de-normalised floating point numbers).
- The error approximating f with $\tilde{f}$ (Lemma 5.9.2) is $< 8.1 \times 10^{-211}$.
- The up-sampling error (Lemmas 5.10.1 and 5.10.3) is $< 5.0 \times 10^{-41}$.

We note that the relatively large up-sampling error is tolerated for computational reasons and can be improved easily by taking more terms in the few exceptional cases.

Chapter 6

Computing $\pi(x)$ Analytically

We turn now to an example application that relies on the ability to compute zeros of Riemann's ζ function efficiently and to high precision, that of computing the prime counting function, $\pi(x)$, analytically.

6.1 History and Background

Computing $\pi(x)$, the number of primes $\leq x$, has a long history. The current (unconditional) world record is $\pi\left(10^{23}\right)$ [63] although a figure for $\pi(10^{24})$ conditional on the Riemann hypothesis has recently been announced by Buethe et al. [18].

The earliest methods involved enumerating the primes less than x and then counting them. However, by the prime number theorem, the number of primes $\leq x$ is asymptotic to $\frac{x}{\log x}$ so even the counting stage is not tractable for large x.

In the 1870's, a new combinatorial method was described by Meissel who used it to compute $\pi(10^9)$ (incorrectly, he was out by 56) which was subsequently improved by Lehmer [53] who used it to compute $\pi(10^{10})$ (incorrectly, but only by 1 this time, and only because he considered 1 to be prime [53]). In 1972 Bohmam [10] computed $\pi(10^{13})$ (incorrectly, out by 941), followed by Lagarias, Miller and Odlyzko [50] who got to $\pi(10^{16})$. The baton then passed

to Deléglise and Rivat [28] who reached $\pi(10^{18})$, then to Gourdon with his distributed implementation [35] who reached $\pi(4 \times 10^{22})$. The unconditional world record mentioned earlier is due to Tomás Oliveira e Silva.

The Meissel method in its latest incarnation has time complexity $\mathcal{O}\left(\frac{x^{2/3}}{\log^2 x}\right)$. Gourdon actually produced a figure for $\pi(10^{23})$ but internal checks revealed an error which could not be isolated. It was about 6 years before the correct figure was computed and this, coupled with the lack of results since 2007 might lead one to suspect that combinatorial methods have reached the limits of their applicability.

However, as long ago as 1987, Lagarias and Odlyzko [49] identified an alternative, analytic approach to this problem and the following sections describe our (unconditional) implementation of that algorithm.

6.2 Derivation of the Analytic Algorithm

6.2.1 Riemann's Explicit Formula

We define

$$\pi^*(x) := \frac{1}{2}\left[\sum_{p^m<x} \frac{1}{m} + \sum_{p^m \leq x} \frac{1}{m}\right].$$

Now, according to Edwards [30], the "main result" of Riemann's 1859 paper [70] was his explicit formula. This states, for $x > 1$,

$$\pi^*(x) = \mathrm{Ei}(\log x) - \sum_{\rho} \mathrm{Ei}(\rho \log x) - \log(2) + \int_x^\infty \frac{\mathrm{d}t}{t(t^2-1)\log(t)}. \qquad (6.2.1)$$

(We note that Edwards uses $\mathrm{Li}(x)$ and $\mathrm{Li}\,(x^\rho)$ respectively in place of the Ei terms. We prefer this formulation to avoid ambiguity in the case of $\mathrm{Li}\,(x^\rho)$ in terms of which branch of the logarithm to take.)

The ρ in the sum are the non-trivial zeros of the Riemann Zeta function, taken in order of increasing absolute imaginary part.

We note $\pi(x)$ can be recovered from $\pi^*(x)$ cheaply (by Möbius inversion) but even so the conditional (and slow) convergence of the sum over ρ renders

this equation useless for computational purposes. Riesel and Göhl describe some computations using this formula and the first 29 pairs of zeros in [71].

6.2.2 The Lagarias and Odlyzko Algorithm

In their 1987 paper [49] Lagarias and Odlyzko suggested that Riemann's explicit formula could be modified to render a computationally efficient, analytic algorithm for $\pi^*(x)$ and hence $\pi(x)$. They go back to Riemann's original derivation of 6.2.1 which starts with, for $\Re s > 1$,

$$\frac{\log \zeta(s)}{s} = \int_0^\infty \pi^*(x) x^{-s} \frac{\mathrm{d}x}{x}.$$

Now since $\pi^*(x)$ is piecewise continuous on $\mathbb{R}_{>0}$ and defined to take the value halfway between the limit values at each jump discontinuity, and since the integral converges absolutely for all $\Re s > 1$, we can apply the Mellin Inversion Theorem to obtain, for $\sigma > 1$,

$$\pi^*(x) = \frac{1}{2\pi i} \int_{\sigma - i\infty}^{\sigma + i\infty} \log \zeta(s) x^s \frac{\mathrm{d}s}{s}.$$

Riemann's explicit formula is the result of evaluating this integral.

At this point, Lagarias and Odlyzko introduce a "suitable" Mellin transform pair $\phi(t)$ and $\hat{\phi}(s)$ and derive

$$\pi^*(x) = \frac{1}{2\pi i} \int_{\sigma - i\infty}^{\sigma + i\infty} \log \zeta(s) \hat{\phi}(s) \, \mathrm{d}s + \sum_{p^m} \frac{1}{m} \left[\chi_x(p^m) - \phi(p^m) \right]$$

where $\chi_x(t)$ is the characteristic function of x, defined by

$$\chi_x(t) := \begin{cases} 1 & t < x \\ 1/2 & t = x. \\ 0 & t > x \end{cases}$$

We note that taking $\hat{\phi}(s) = \frac{x^s}{s}$ makes $\phi(t) = \chi_x(t)$ and we recover Riemann's explicit formula.

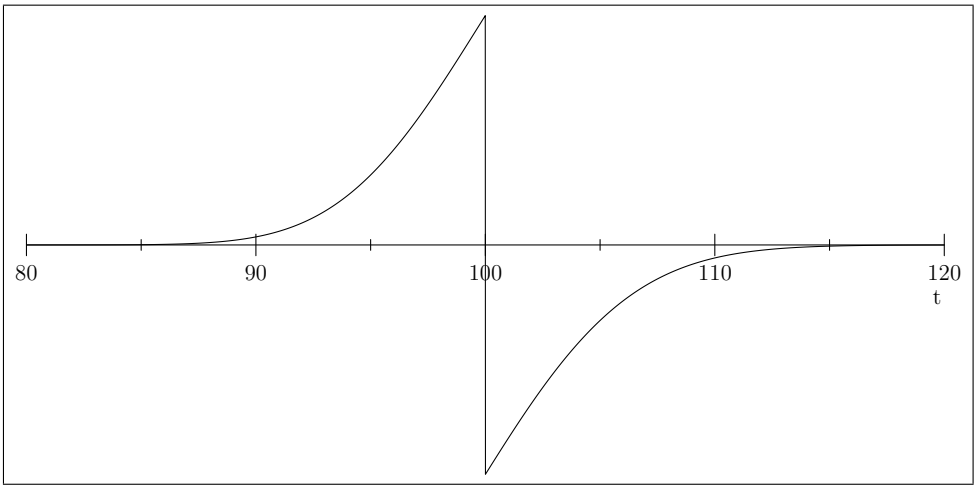

Figure 6.1: $\chi_x(t) - \phi(t)$ for $x = 100$ and $\lambda = \frac{1}{20}$

6.2.3 Galway's Analysis

In his PhD thesis [33] Galway investigated the proposed algorithm and suggested using the Mellin transform pair

$$\hat{\phi}(s) := \frac{x^s}{s} \exp\left(\frac{\lambda^2 s^2}{2}\right)$$

and

$$\phi(t) := \frac{1}{2}\text{erfc}\left(\frac{\log\left(\frac{t}{x}\right)}{\sqrt{2}\lambda}\right).$$

Here erfc the complementary error function

$$\text{erfc}(x) := \frac{2}{\sqrt{\pi}} \int_x^\infty \exp\left(-t^2\right)\, \mathrm{d}t$$

and λ is a positive real parameter used to balance the convergence of the integral with that of the prime sieve.

Galway showed that ϕ and $\hat{\phi}$ as defined are indeed "suitable". Further the Fourier uncertainty principle (see Appendix 1 of [89]) informally stated says that there is a lower limit to the product of the widths of a function and its Fourier transform. Since the Gaussian is the only function that achieves this lower limit, the pair suggested by Galway are in some sense at least optimal. An example graph of the prime sieving element is given as Figure 6.1

Like Lagarias and Odlyzko before him, Galway proposed computing the integral numerically. This poses some difficulties, notably the size of x^σ once we move past $s = 1$. We therefore adopt a different approach, one that more closely follows Riemann's 1859 paper.

6.3 Evaluating the Integral

6.3.1 A Contour Integral

We wish to evaluate the integral $\frac{1}{2\pi i}\int_{\sigma-i\infty}^{\sigma+i\infty} \hat{\phi}(s)\log\zeta(s)\,\mathrm{d}s$.

Before proceeding with the integral, we need a couple of lemmas.

Lemma 6.3.1. *The "Round the Pole" lemma. Given a function f with a simple pole at α with residue R, let Γ be the semicircular contour anticlockwise from $\alpha+\epsilon$ to $\alpha-\epsilon$. Then*

$$\lim_{\epsilon\to 0^+}\int_\Gamma f(z)\,\mathrm{d}z = \pi i R.$$

Proof. See page 29 of [82]. □

Lemma 6.3.2. *Let ϵ be small and positive. Then*

$$\lim_{\epsilon\to 0^+}(\log\zeta(1+\epsilon) - \log(-\zeta(1-\epsilon))) = 0.$$

Proof.

$$\begin{aligned}&\lim_{\epsilon\to 0^+}(\log\zeta(1+\epsilon) - \log(-\zeta(1-\epsilon)))\\ &= \lim_{\epsilon\to 0^+}\log\frac{\zeta(1+\epsilon)}{-\zeta(1-\epsilon)}\\ &= \lim_{\epsilon\to 0^+}\log\frac{1/\epsilon+\mathcal{O}(1)}{1/\epsilon+\mathcal{O}(1)} = 0\end{aligned}$$

□

Lemma 6.3.3. *There exists a sequence of $T_j\to\infty$ such that for any $\sigma\in[-1,2]$ we have for $s=\sigma+iT_j$*

$$\frac{\zeta'}{\zeta}(s) = \mathcal{O}(\log^2 T_j).$$

Proof. Referring to Davenport [26], for any zero $\beta+i\gamma$ of ζ with γ large, there are $\mathcal{O}(\log\gamma)$ zeros with imaginary part $\in[\gamma-1,\gamma+1]$ (Corollary (a), page 99). Therefore we can select a T_j within $\mathcal{O}(1)$ of γ such that T_j differs from the imaginary part of any zero by $\gg\frac{1}{\log T_j}$. By (4) on page 99 we have for $\sigma\in[-1,2]$

$$\left|\frac{\zeta'}{\zeta}\left(\frac{1}{2}+iT_j\right)\right|=\left|{\sum_{\rho}}'\frac{1}{\frac{1}{2}+iT_j-\rho}+\mathcal{O}(\log T_j)\right|$$

where the sum is taken over zeros with imaginary part $\in[T_j-1,T_j+1]$. There are $\mathcal{O}(\log T_j)$ such zeros, each one making a contribution to the sum limited by $\mathcal{O}(\log T_j)$ and the result follows. □

Lemma 6.3.4. *For $t\in\mathbb{R}$*

$$|\log(-\zeta(-1+it))|\leq 5+t^2.$$

Proof. By the functional equation

$$\log(-\zeta(-1+it))=\log\zeta(2-it)+\log\Gamma\left(\frac{2-it}{2}\right)-\log\Gamma\left(\frac{-1+it}{2}\right)+\left(\frac{3}{2}-it\right)\log\pi.$$

We then use $\left(\frac{-1+it}{2}\right)\Gamma\left(\frac{-1+it}{2}\right)=\Gamma\left(\frac{1+it}{2}\right)$ so we can apply Stirling's approximation. Also, for $\Re s>1$ we have

$$|\log\zeta(s)|=\left|\sum_{n=2}^{\infty}\frac{\Lambda(n)}{\log(n)}\frac{1}{n^s}\right|\leq\sum_{n=2}^{\infty}\frac{\Lambda(n)}{\log(n)}\frac{1}{n^{\Re s}}=\log\zeta(\Re s).$$

□

Lemma 6.3.5.

$$\left|\frac{1}{2\pi i}\int_{-1-i\infty}^{-1+i\infty}\log(-\zeta(s))\widehat{\phi}(s)\,\mathrm{d}s\right|\leq\frac{\exp\left(\frac{\lambda^2}{2}\right)}{2\pi x\lambda}\left(5\sqrt{2\pi}+\frac{2}{\lambda}\right).$$

Proof. We use Lemma 6.3.4 and take absolute values, majorising with

$$\frac{\exp\left(\frac{\lambda^2}{2}\right)}{2\pi x}\left[5\int_{-\infty}^{\infty}\exp\left(\frac{-\lambda^2t^2}{2}\right)\,\mathrm{d}t+\int_{-\infty}^{\infty}|t|\exp\left(\frac{-\lambda^2t^2}{2}\right)\,\mathrm{d}t\right]$$

where both integrals can be evaluated. □

Lemma 6.3.6. *Let $\widehat{\Phi}(s)$ be the unique holomorphic function $\widehat{\Phi} : \mathbb{C} \setminus \mathbb{R}_{\leq 0} \to \mathbb{C}$ such that*

- $\widehat{\Phi}'(s) = \widehat{\phi}(s)$ *and*
- $\lim_{t\to\infty} \left[\widehat{\Phi}(\sigma + it) + \widehat{\Phi}(\sigma - it)\right] = 0$ *for any fixed real σ.*

Then

1. $\widehat{\Phi}(s) - \log s$ *extends to an entire function,*
2. $\lim_{t\to\infty} \widehat{\Phi}(\sigma + it) = C$ *is purely imaginary and*
3. $\widehat{\Phi}(\sigma \pm it) \mp C$ *is rapidly decreasing as $t \to \infty$.*

Proof. To show (1) we define for $s \notin \mathbb{R}_{\leq}0$

$$F(s) = \int_1^s \widehat{\phi}(z)\,\mathrm{d}z$$

where the contour of integration is the straight line from 1 to s. We then define

$$\widehat{\Phi}(s) := \lim_{T\to\infty} \left[F(s) - \frac{F(1+iT) + F(1-iT)}{2}\right]$$

and we have

$$F(s) - \log s = \int_1^s \left(\widehat{\phi}(z) - \frac{1}{z}\right)\,\mathrm{d}z.$$

To show (2) we observe that $\widehat{\Phi}(\overline{s}) = \overline{\widehat{\Phi}(s)}$ and from the definition we have $C + \overline{C} = 0$. To show (3), we take $T > 0$ and we have

$$|\widehat{\Phi}(\sigma \pm iT) \mp C| \leq \frac{x^\sigma}{T} \exp\left(\frac{\sigma^2\lambda^2}{2}\right) \int_T^\infty \exp\left(\frac{-\lambda^2 t^2}{2}\right)\,\mathrm{d}t.$$

□

Theorem 6.3.7. *Let $\widehat{\Phi}(s)$ and C be as defined in Lemma 6.3.6. Then*

$$\frac{1}{2\pi i}\int_{2-i\infty}^{2+i\infty} \widehat{\phi}(s)\log\zeta(s)\,\mathrm{d}s = \widehat{\Phi}(1) - \sum_\rho \Re\widehat{\Phi}(\rho) - \log(2) + \frac{1}{2\pi i}\int_{-1-i\infty}^{-1+i\infty} \widehat{\phi}(s)\log(-\zeta(s))\,\mathrm{d}s.$$

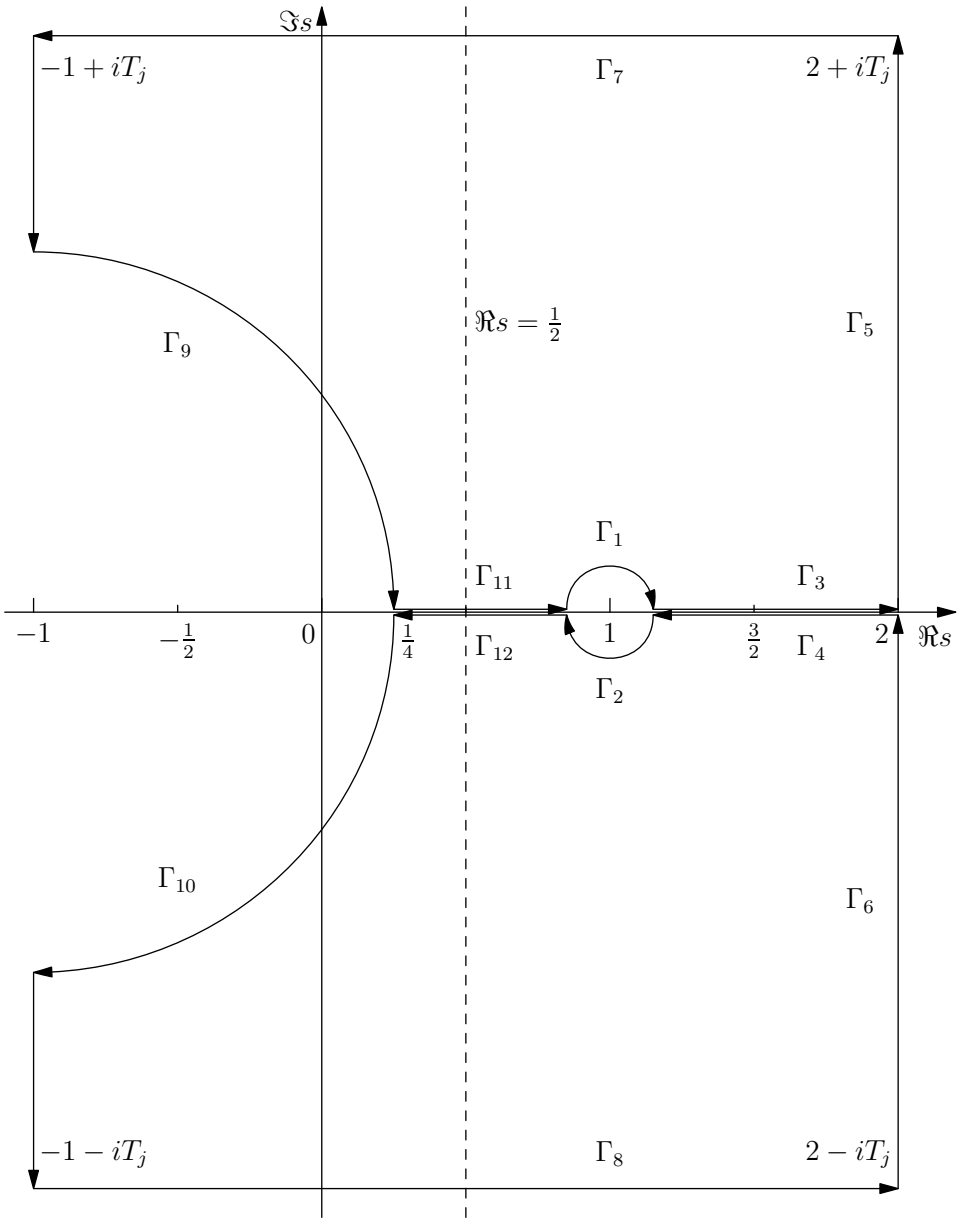

Figure 6.2: Contours to evaluate $\frac{1}{2\pi i}\int_{2-i\infty}^{2+i\infty}\hat{\phi}(s)\log\zeta(s)\,\mathrm{d}s$

Proof. We will refer to the contours represented in Figure 6.2.

These contours are

- Γ_1 - the semi-circle clockwise from $1-\epsilon$ to $1+\epsilon$ for ϵ small and positive.
- Γ_2 - the semi-circle clockwise from $1+\epsilon$ to $1-\epsilon$.
- Γ_3 - the horizontal line from $1+\epsilon$ to 2.
- Γ_4 - the horizontal line from 2 to $1+\epsilon$.
- Γ_5 - the vertical line from 2 to $2+iT_j$, T_j not the ordinate of a zero of ζ.
- Γ_6 - the vertical line from $2-iT_j$ to 2.
- Γ_7 - the horizontal line from $2+iT_j$ to $-1+iT_j$.
- Γ_8 - the horizontal line from $-1-iT_j$ to $2-iT_j$.
- Γ_9 - the vertical line from $-1+iT_j$ to $-1+\frac{5}{4}i$, followed by the clockwise circular arc centred at -1 to $\frac{1}{4}$.
- Γ_{10} - the clockwise circular arc centred at -1 from $\frac{1}{4}$ to $-1-\frac{5}{4}i$, followed by the vertical line to $-1-iT_j$.
- Γ_{11} - the horizontal line from $\frac{1}{4}$ to $1-\epsilon$.
- Γ_{12} - the horizontal line from $1-\epsilon$ to $\frac{1}{4}$.

We consider the integrals

$$\frac{1}{2\pi i}\int(\widehat{\Phi}(s)-C)\frac{\zeta'(s)}{\zeta(s)}\,\mathrm{d}s \tag{6.3.1}$$

for the contours Γ_1, Γ_3, Γ_5, Γ_7 Γ_9 and Γ_{11} in the upper half plane and

$$\frac{1}{2\pi i}\int(\widehat{\Phi}(s)+C)\frac{\zeta'(s)}{\zeta(s)}\,\mathrm{d}s \tag{6.3.2}$$

for Γ_2, Γ_4, Γ_6, Γ_8 Γ_{10} and Γ_{12} in the lower half.

We denote the integrals in Equations 6.3.1 or 6.3.2 as appropriate along Γ_n by I_n and proceed as follows.

For I_5 and I_6 we get

$$\begin{aligned}\lim_{j\to\infty}(I_5+I_6) &= \lim_{j\to\infty}\frac{1}{2\pi i}\left[\int_{\Gamma_5}\left(\widehat{\Phi}(s)-C\right)\frac{\zeta'}{\zeta}(s)\,\mathrm{d}s+\int_{\Gamma_6}\left(\widehat{\Phi}(s)+C\right)\frac{\zeta'}{\zeta}(s)\,\mathrm{d}s\right]\\ &= \lim_{j\to\infty}\frac{1}{2\pi i}\left[\left(\widehat{\Phi}(s)-C\right)\log\zeta(s)\Big|_2^{2+iT_j}+\left(\widehat{\Phi}(s)+C\right)\log\zeta(s)\Big|_{2-iT_j}^{2}\right]\\ &\quad-\frac{1}{2\pi i}\left[\int_{\Gamma_{5,6}}\widehat{\phi}(s)\log\zeta(s)\,\mathrm{d}s\right]\\ &= \frac{1}{2\pi i}\left[2C\log\zeta(2)-\int_{2-i\infty}^{2+i\infty}\widehat{\phi}(s)\log\zeta(s)\,\mathrm{d}s\right].\end{aligned}$$

Considering the contours Γ_7 and Γ_8, we use Lemma 6.3.3 and the Gaussian decay of $\widehat{\Phi}(s)\pm C$ from Lemma 6.3.6 to conclude

$$\begin{aligned}\lim_{j\to\infty}(I_7+I_8) &= \lim_{j\to\infty}\frac{1}{2\pi i}\left[\int_{\Gamma_7}\left(\widehat{\Phi}(s)-C\right)\frac{\zeta'}{\zeta}(s)\,\mathrm{d}s+\int_{\Gamma_8}\left(\widehat{\Phi}(s)+C\right)\frac{\zeta'}{\zeta}(s)\,\mathrm{d}s\right]\\ &= 0.\end{aligned}$$

Considering I_9 and I_{10} we have

$$\begin{aligned}\lim_{j\to\infty}(I_9+I_{10}) &= \lim_{j\to\infty}\frac{1}{2\pi i}\left[\int_{\Gamma_9}\left(\widehat{\Phi}(s)-C\right)\frac{\zeta'}{\zeta}(s)\,\mathrm{d}s+\int_{\Gamma_{10}}\left(\widehat{\Phi}(s)+C\right)\frac{\zeta'}{\zeta}(s)\,\mathrm{d}s\right]\\ &= \lim_{j\to\infty}\frac{1}{2\pi i}\left[\left(\widehat{\Phi}(s)-C\right)\log(-\zeta(s))\Big|_{-1+iT_j}^{1/4}\right.\\ &\qquad\left.+\left(\widehat{\Phi}(s)+C\right)\log(-\zeta(s))\Big|_{1/4}^{-1-iT_j}\right]\\ &\quad-\frac{1}{2\pi i}\left[\int_{\Gamma_9}\widehat{\phi}(s)\log(-\zeta(s))\,\mathrm{d}s+\int_{\Gamma_{10}}\widehat{\phi}(s)\log(-\zeta(s))\,\mathrm{d}s\right]\\ &= -\frac{1}{2\pi i}\left[\int_{\Gamma_9,\Gamma_{10}}\widehat{\phi}(s)\log(-\zeta(s))\,\mathrm{d}s+2C\log\left(-\zeta\left(\frac{1}{4}\right)\right)\right]\end{aligned}$$

where the contour of integration is Γ_9 followed by Γ_{10}. Convergence of this integral is due to Lemma 6.3.5 and the zero free region of $\zeta(s)$ with $|s|\leq\frac{1}{2}$.

For I_{11} and I_{12} we have

$$\begin{aligned} I_{11}+I_{12} &= \frac{1}{2\pi i}\left[\int_{\Gamma_{11}}\left(\widehat{\Phi}(s)-C\right)\frac{\zeta'}{\zeta}(s)\,\mathrm{d}s+\int_{\Gamma_{12}}\left(\widehat{\Phi}(s)+C\right)\frac{\zeta'}{\zeta}(s)\,\mathrm{d}s\right] \\ &= \frac{1}{2\pi i}\left[\left(\widehat{\Phi}(s)-C\right)\log(-\zeta(s))\Big|_{1/4}^{1-\epsilon}+\left(\widehat{\Phi}(s)+C\right)\log(-\zeta(s))\Big|_{1-\epsilon}^{1/4}\right] \\ &\quad -\frac{1}{2\pi i}\left[\int_{\Gamma_{11}}\widehat{\phi}(s)\log(-\zeta(s))\,\mathrm{d}s+\int_{\Gamma_{12}}\widehat{\phi}(s)\log(-\zeta(s))\,\mathrm{d}s\right] \\ &= \frac{1}{2\pi i}\left[2C\log\left(-\zeta\left(\frac{1}{4}\right)\right)-2C\log(-\zeta(1-\epsilon))\right]. \end{aligned}$$

For I_1 and I_2 we find

$$\begin{aligned} I_1+I_2 &= \frac{1}{2\pi i}\left[\int_{\Gamma_1}\left(\widehat{\Phi}(s)-C\right)\frac{\zeta'}{\zeta}(s)\,\mathrm{d}s+\int_{\Gamma_2}\left(\widehat{\Phi}(s)+C\right)\frac{\zeta'}{\zeta}(s)\,\mathrm{d}s\right] \\ &= \frac{1}{2\pi i}\left[\oint\widehat{\Phi}(s)\frac{\zeta'(s)}{\zeta(s)}\,\mathrm{d}s-C\int_{\Gamma_1}\frac{\zeta'}{\zeta}(s)\,\mathrm{d}s+C\int_{\Gamma_2}\frac{\zeta'}{\zeta}(s)\,\mathrm{d}s\right] \\ &= \widehat{\Phi}(1)-\frac{C}{2\pi i}\left[\int_{\Gamma_1}\frac{\zeta'}{\zeta}(s)\,\mathrm{d}s-\int_{\Gamma_2}\frac{\zeta'}{\zeta}(s)\,\mathrm{d}s\right] \end{aligned}$$

by Cauchy's Theorem since the residue of $\frac{\zeta'}{\zeta}$ at 1 is -1.

Finally, for I_3 and I_4 we get

$$\begin{aligned} I_3+I_4 &= \frac{1}{2\pi i}\left[\int_{\Gamma_3}\left(\widehat{\Phi}(s)-C\right)\frac{\zeta'}{\zeta}(s)\,\mathrm{d}s+\int_{\Gamma_4}\left(\widehat{\Phi}(s)+C\right)\frac{\zeta'}{\zeta}(s)\,\mathrm{d}s\right] \\ &= \frac{1}{2\pi i}\left[\left(\widehat{\Phi}(s)-C\right)\log\zeta(s)\Big|_{1+\epsilon}^{2}+\left(\widehat{\Phi}(s)+C\right)\log\zeta(s)\Big|_{2}^{1+\epsilon}\right] \\ &\quad -\frac{1}{2\pi i}\left[\int_{\Gamma_3}\widehat{\phi}(s)\log\zeta(s)\,\mathrm{d}s+\int_{\Gamma_4}\widehat{\phi}(s)\log\zeta(s)\,\mathrm{d}s\right] \\ &= \frac{1}{2\pi i}\left[2C\log\zeta(1+\epsilon)-2C\log\zeta(2)\right]. \end{aligned}$$

Now by Cauchy's Theorem again, $\lim_{j\to\infty}\sum_{k=1}^{12}I_k=\sum_{\rho}\widehat{\Phi}(\rho)$ so we have

$$\sum_{\rho} \widehat{\Phi}(\rho) = \widehat{\Phi}(1) - \frac{1}{2\pi i}\left[\int_{2-i\infty}^{2+i\infty} \widehat{\phi}(s)\log\zeta(s)\,\mathrm{d}s + \int_{\Gamma_9,\Gamma_{10}} \widehat{\phi}(s)\log(-\zeta(s))\,\mathrm{d}s\right]$$
$$+ \frac{C}{\pi i}\left[\log\zeta(1+\epsilon) - \log(-\zeta(1-\epsilon))\right]$$
$$- \frac{C}{2\pi i}\left[\int_{\Gamma_1} \frac{\zeta'(s)}{\zeta(s)}\,\mathrm{d}s - \int_{\Gamma_2} \frac{\zeta'(s)}{\zeta(s)}\,\mathrm{d}s\right].$$

Now the result follows from taking the limit as $\epsilon \to 0^+$ by Lemmas 6.3.2 and 6.3.1 and then straightening the line of integration of the second integral to $\Re s = -1$. This introduces a contribution of $\log -\zeta(0) = -\log 2$ from the pole of $\widehat{\phi}(s)$ at $s = 0$ with residue 1. □

6.3.2 Computing $\widehat{\Phi}\left(\frac{1}{2} + it\right)$

We now need an effective method for computing $\widehat{\Phi}$. The following lemma is our starting point.

Lemma 6.3.8. *For $\Re s_0 \neq 0$ and $h \in \mathbb{R}$*

$$\hat{\phi}(s_0 + ih) = \hat{\phi}(s_0)\exp\left(ih(s_0\lambda^2 + \log(x))\right)\frac{\exp\left(\frac{-\lambda^2h^2}{2}\right)}{1 + \frac{ih}{s_0}}.$$

Proof. We start with

$$\hat{\phi}(s_0 + ih) = \frac{\exp\left(\frac{\lambda^2(s_0+ih)^2}{2}\right)x^{s_0+ih}}{s_0 + ih}$$

and rearrange to get

$$\frac{\exp\left(\frac{\lambda^2 s_0^2}{2}\right)x^{s_0}}{s_0}\frac{\exp\left(ih(s_0\lambda^2 + \log(x))\right)\exp\left(\frac{-\lambda^2h^2}{2}\right)}{1 + \frac{ih}{s_0}}.$$

□

Lemma 6.3.9. *Let $N \in 2\mathbb{Z}_{>0}$, $\lambda, h > 0$ and $\lambda h < 1$. Then*

$$\exp\left(\frac{-\lambda^2h^2}{2}\right) = \sum_{n=0}^{\frac{N}{2}} \frac{(-1)^n}{n!}\left(\frac{\lambda^2h^2}{2}\right)^n + E_A,$$

where

$$|E_A| \leq \frac{1}{\left(\frac{N}{2}\right)!}\left(\frac{\lambda^2 h^2}{2}\right)^{\frac{N}{2}}.$$

Proof. This function is entire and the restriction on λh makes the terms alternating in sign and decreasing. □

Lemma 6.3.10. *Let $N \in \mathbb{Z}_{>0}$ and $|h| < |s_0|$. Then*

$$\left(1 + \frac{ih}{s_0}\right)^{-1} = \sum_{n=0}^{N}\left(\frac{-ih}{s_0}\right)^n + E_B,$$

where if $R = \left|\frac{h}{s_0}\right|$ we have

$$|E_B| \leq \frac{R^N}{1-R}.$$

Proof. This function is analytic on the open disk $|h| < |s_0|$ and the missing terms form a geometric series. □

We can now fix some $N \in 2\mathbb{Z}_{>0}$ and multiply these two (degree N) polynomials to yield a single (degree $2N$) polynomial in h which we can integrate against $\exp(ih(\lambda^2 + \log(x)))$ analytically. We note in passing that the polynomial multiplication can be achieved efficiently by recourse to the Convolution Theorem but in practice the limited number of terms required mean that the trivial $\mathcal{O}(N^2)$ algorithm suffices. We also mention that a more efficient version of the Taylor approximation would be obtained if we computed it as a single expansion, rather than splitting it into two. However, this approach sufficed for our purposes.

We must consider the error introduced by truncating these Taylor expansions and the following lemma addresses this.

Lemma 6.3.11. *Let $\lambda, x > 0$, $H \in (0, |s_0|)$, $\Re s_0 \neq 0$, $R = \left|\frac{H}{s_0}\right|$ and $\hat{\phi}_N(s)$ denote the approximation to $\hat{\phi}(s)$ resulting from truncating 6.3.9 at $\frac{N}{2}$ and 6.3.10 at N. Then*

$$\left|\int_{-H}^{H} \hat{\phi}(s_0 + ih) - \hat{\phi}_N(s_0 + ih)\,\mathrm{d}h\right| \leq 2H\left|\hat{\phi}(s_0)\right|\left(|E_A|\,|E_B| + |E_B| + \frac{|E_A|}{1-R}\right).$$

Proof. We multiply out the expressions given in Lemmas 6.3.9 and 6.3.10 and estimate the resulting integrals using absolute values. □

6.3.3 More Error Bounds

We now provide rigorous bounds for the other sources of error that will not be handled through interval arithmetic.

For $t > 0$, t not the imaginary part of a zero of ζ, define $N(t)$ to be the number of zeros of $\zeta(s)$ in the critical strip with $\Im s \in [0, t]$.

Lemma 6.3.12. *Let $t \geq 2$. Then*

$$\left| N(t) - \frac{t}{2\pi} \log\left(\frac{t}{2\pi\,\mathrm{e}}\right) - \frac{7}{8} \right| < 0.137 \log(t) + 0.443 \log\log(t) + 1.588.$$

Proof. See [73] Theorem 19. □

Lemma 6.3.13. *For $x, T, \lambda > 0$ and $\sigma \in [0, 1]$ define*

$$B(\sigma, T) := \exp\left(\frac{\lambda^2(1-T^2)}{2}\right)\left[\frac{x^\sigma}{T\log x} + \frac{1}{\lambda^2 T^2 x}\right].$$

Then

$$\left|\Re\widehat{\Phi}\left(\sigma + iT\right)\right| \leq B(\sigma, T).$$

Proof. We integrate along the contour running vertically down from $-1 + i\infty$ to $-1 + iT$, then right to $\sigma + iT$. For the horizontal contour we have

$$\begin{aligned}\left|\int_{-1}^{\sigma} \frac{\exp\left(\frac{\lambda^2(u+iT)^2}{2}\right)}{u+iT} x^{u+iT}\,\mathrm{d}u\right| &\leq \frac{\exp\left(\frac{\lambda^2(1-T^2)}{2}\right)}{T}\int_{-1}^{\sigma} x^u\,\mathrm{d}u \\ &< \frac{\exp\left(\frac{\lambda^2(1-T^2)}{2}\right)}{T\log x} x^\sigma.\end{aligned}$$

For the vertical contour we have

$$\begin{aligned}\left|\int_{\infty}^{T} \frac{\exp\left(\frac{\lambda^2(-1+it)^2}{2}\right)}{-1+it} x^{-1+it}\,\mathrm{d}t\right| &\leq x^{-1}\exp\left(\frac{\lambda^2}{2}\right)\int_{\infty}^{T}\frac{\exp\left(\frac{-\lambda^2 t^2}{2}\right)}{t}\,\mathrm{d}t \\ &< \frac{\exp\left(\frac{\lambda^2}{2}\right)}{xT^2}\int_{\infty}^{T} t\exp\left(\frac{-\lambda^2 t^2}{2}\right)\,\mathrm{d}t \\ &= \frac{\exp\left(\frac{\lambda^2(1-T^2)}{2}\right)}{\lambda^2 T^2 x}.\end{aligned}$$

□

Lemma 6.3.14. *Let $T > 0$, $\sigma \in [0,1]$ and α_T be such $t^{\alpha_T} \geq N(t)$ for all $t \geq T$. Then*

$$\sum_{\Im\rho\geq T} \Re\widehat{\Phi}(\rho) \leq \exp\left(\frac{\lambda^2(1-T^2)}{2}\right)\left[\frac{x^\sigma}{T\log x}+\frac{1}{\lambda^2T^2x}\right]\left[\frac{\lambda^2T^2+2}{\lambda^2T^{2-\alpha_T}}-N(T)\right].$$

Proof. Referring to Lemma 6.3.13 and writing $k_\sigma := \exp\left(\frac{\lambda^2}{2}\right)\left[\frac{x^\sigma}{T\log x}+\frac{1}{\lambda^2T^2x}\right]$, we can majorise the sum with the Stieltjes integral

$$\int_{T}^{\infty} k_\sigma \exp\left(\frac{-\lambda^2t^2}{2}\right)\,\mathrm{d}N(t).$$

We now integrate by parts and majorise $N(t)$ with t^{α_T} to obtain

$$\begin{aligned}\sum_{\Im\rho>T} B(\sigma,\Im\rho) &\leq -k_\sigma\exp\left(\frac{-\lambda^2T^2}{2}\right)N(T) - \frac{k_\sigma}{T^{2-\alpha_T}}\int_{T}^{\infty}\lambda^2t^3\exp\left(\frac{-\lambda^2t^2}{2}\right)\,\mathrm{d}t \\ &= k_\sigma\left[\frac{\lambda^2T^2+2}{\lambda^2T^{2-\alpha_T}}\exp\left(\frac{-\lambda^2T^2}{2}\right)-\exp\left(\frac{-\lambda^2T^2}{2}\right)N(T)\right].\end{aligned}$$

□

We note that the α_T referred to above can be computed using Lemma 6.3.12.

We now consider the error introduced by truncating our sum over the zeros of ζ. Let T_1 be height below which we find and use all the zeros, and T_2 the height to which we know RH to hold. For our computations, we take the results from the Zetagrid calculations [88], which showed that at least the first

900 000 000 000 zeros lie on the 1/2 line, which equates to a value for T_2 of at least 2.419×10^{11}. Zetagrid used Rieman-Siegel and its developers took great care over error management so we feel justified in relying on its conclusions. We note, however, that setting $T_1 = T_2$ in the following (i.e. only believing RH up to the height we ourselves have verified) would require us to compute to height about $\sqrt{2}T_1$ to keep the sieve width constant.

Lemma 6.3.15. *Let E_1 be real part of the error introduced by ignoring zeros with imaginary part of absolute value $\in [T_1, T_2]$ (whose real parts are all known to be $\frac{1}{2}$). Then*

$$|E_1| \leq 2 \exp\left(\frac{\lambda^2(1-T_1^2)}{2}\right)\left[\frac{\sqrt{x}}{T_1 \log x} + \frac{1}{\lambda^2 T_1^2 x}\right]\left[\frac{\lambda^2 T_1^2 + 2}{\lambda^2 T_1^{2-\alpha_{T_1}}} - N(T_1)\right].$$

Proof. We apply Lemma 6.3.14 with $\sigma = \frac{1}{2}$ and introduce a factor of 2 for the zeros with negative imaginary part. □

We note this bound includes all the zeros with imaginary part $> T_2$ but their contribution will be negligible.

Lemma 6.3.16. *Let E_2 be the real part of the error introduced by omitting the zeros with imaginary part $\notin [-T_2, T_2]$ from the main sum, (which do not necessarily have real part $= \frac{1}{2}$). Then*

$$|E_2| \leq \exp\left(\frac{\lambda^2(1-T_2^2)}{2}\right)\left[\frac{x+1}{T_2 \log x} + \frac{2}{\lambda^2 T_2^2 x}\right]\left[\frac{\lambda^2 T_2^2 + 2}{\lambda^2 T_2^{2-\alpha_{T_2}}} - N(T_2)\right].$$

Proof. We pair each ρ with $1 - \overline{\rho}$ and take the worst case when one of the zeros has real part 1 and the other 0. The result then follows from Lemma 6.3.14. □

6.4 The Sum Over Prime Powers

To enable us to compute the contribution to $\pi^*(x)$ from the prime powers, we need the following:

- A rigorous bound for the error truncating the sieve to a finite window.
- A means of enumerating the primes within that window
- A means of computing $\phi(p)$ over those primes

We note that we will only consider the primes themselves. The task for prime squares and higher powers is trivial by comparison.

We dispense with the error bound immediately whereas enumerating the primes and computing $\phi(p)$ efficiently will be the subject of the next two sections.

Lemma 6.4.1. *Let* $\epsilon \in (0, \lambda x]$ *and* $\tau := \lambda\left(\sqrt{2\log\left(\frac{\lambda x}{\epsilon}\right)} + 1\right)$. *Then*

$$\left|\sum_{p^m} \frac{1}{m}\left(\chi_x\left(p^m\right) - \phi\left(p^m\right)\right) - \sum_{p^m \in [x\,\mathrm{e}^{-\tau}, x\,\mathrm{e}^{\tau}]} \frac{1}{m}\left(\chi_x\left(p^m\right) - \phi\left(p^m\right)\right)\right| \leq \epsilon.$$

Proof. See Galway [33] Theorem 3.10. □

6.4.1 Enumerating Primes in Wide, High Intervals

To compute $\pi(x)$ with $x = 10^{23}$ with access to the non-trivial zeros of ζ to height $\leq 1.1 \times 10^{10}$ or so required us to enumerate the primes in a window of width $\approx 1.1 \times 10^{15}$ (see section 6.5.2 for details on the choice of parameters). By the prime number theorem, we would expect to find about 2.1×10^{13} primes.

Two alternative basic techniques present themselves, primality testing and sieving.

6.4.1.1 Primality Tests

One approach would be to quickly sieve out the majority of composites and then to apply a strong primality test to the remainder. Experiments suggest that such methods will not be competitive. For example, using Pari's *isprime* function [5] is orders of magnitude slower at $x = 10^{23}$ than the method we eventually selected.

A related method would be to enumerate the Fermat pseudoprimes, say base 2 and 3, within our window. We would then sieve the segment using small primes, strike out the known Fermat pseudoprimes, then sieve again using the (weak) Fermat primality test. Again initial investigations suggested that this method is not going to compete with sieving at these heights.

Both of these methods have the advantage of being very efficient in terms of space when compared with prime sieves. However, these advantages do not manifest themselves at $x = 10^{23}$, nor would we expect them to at 10^{24}.

6.4.1.2 The Sieve of Eratosthenes

The oldest (B.C.) and simplest prime sieve is the Sieve of Eratosthenes. In pseudo code, the algorithm is:

```
for i:=2 to x
do
   primes[i]:=true
od
for j=4 to x by 2
do
   primes[j]:=false
od
for i:=3 to floor(sqrt(x)) by 2
do
   if primes[i] then
      for j := i*i to N by i
      do
         primes[j]:=false
      od
   fi
od
```

Since for each prime $p \leq \sqrt{x}$ we must cross out $\mathcal{O}(\frac{x}{p})$ multiples of p, the number of operations for the Sieve of Eratosthenes is

$$\sum_{p \leq \sqrt{x}} \frac{x}{p} = \mathcal{O}(x \log \log x)$$

The space required is $\mathcal{O}(x)$. We note that by working modulo K and allowing K to increase slowly with x, we can save a factor of $\log \log x$ in the time complexity but that is not significant in what follows.

The width of the interval we wish to sieve, even for $\pi(10^{23})$, is too large to fit in memory. At 1 bit per integer and working modulo $2 \times 3 \times 5$ we would need 37 Tbytes. We are therefore forced to segment the sieve [6].

Assume in what follows that our sieve interval of width W ends at x and that we segment it into pieces of length S (determined by the memory we have available to us). We must compute the base primes up to $\sqrt{x}$ (not necessarily all at once). Here we have several options:

- Pre-compute all the primes and store them on disk. Read them in as required.
- Use (segmented) Eratosthenes recursively to compute the base primes on demand for each segment.
- Use some other method to compute the primes on demand.

Storing the pre-computed primes to disk is feasible. We store the differences between successive primes divided by 2 as single bytes and for the few cases where the gap between successive primes exceeds 511 we use a simple multi-byte encoding. However, simply reading this 35 Gbyte file takes a comparable amount of time to computing the primes on the fly, and that is before taking account of I/O clashes when multiple cores try to access the same device.

Recursively segmenting Eratosthenes down to $x^{1/3}$ or so (by which point we can compute all the primes in memory) is also competitive. However, using Bernstein's own 64 bit implementation of the asymptotically better Atkin-Bernstein sieve (see below and [7]) outperformed both.

The run time appears to be dominated by the time taken to locate the first multiple of each base prime in the target segment (or to conclude there isn't one), probably because it involves 128 bit division. This contributes $\mathcal{O}\left(\frac{\sqrt{x}}{\log x}\right)$ but, as we will discuss shortly, using the Atkin-Bernstein sieve to compute the primes to $\sqrt{x}$ is already $\mathcal{O}\left(\frac{\sqrt{x}}{\log\log x}\right)$ so the overall time complexity is $\mathcal{O}\left(\frac{W\sqrt{x}}{S\log\log x}\right)$.

6.4.1.3 The Atkin-Bernstein Sieve

With the Sieve of Eratosthenes, we are considering solutions of the reducible binary quadratic form xy. If we use irreducible binary quadratic forms instead, we construct a sieve first described by Atkin and Bernstein [4]. Specifically for square free $n > 3$

- $n \in 1 + 4\mathbb{Z}$ is prime iff $\#\{(x, y) : x > 0, y > 0, 4x^2 + y^2 = n\}$ is odd.
- $n \in 7 + 12\mathbb{Z}$ is prime iff $\#\{(x, y) : x > 0, y > 0, 3x^2 + y^2 = n\}$ is odd.
- $n \in 11 + 12\mathbb{Z}$ is prime iff $\#\{(x, y) : x > y > 0, 3x^2 - y^2 = n\}$ is odd.

Atkin and Bernstein do not claim that this choice of binary quadratic forms is optimal. They go on to show that its time complexity is $\mathcal{O}(x)$ (or $\mathcal{O}\left(\frac{x}{\log\log x}\right)$ if we work modulo K and allow K to increase slowly with x) and $\mathcal{O}(\sqrt{x})$ space.

This sieve segments more readily than the Sieve of Eratosthenes because there is no need to re-compute "small" primes for each segment. The difficulty that arises is a practical one. The n in the congruences given above will all be of a size that exceeds the 64 bit native word size of most modern CPUs. (64 bits equates to about 10^{19}). This means we must use software library routines in place of native hardware instructions and to date we have been unable to produce a version of Atkin-Bernstein more efficient than Eratosthenes at this height.

6.4.1.4 Dissected Sieves

In his thesis [33] Galway introduces a novel sieving method based on Atkin-Bernstein which he referred to as a dissected sieve. His analysis showed it to have time complexity (in our terms) of $\mathcal{O}\left(W\left(1+\frac{x^{1/3}}{S}\right)\right)$ with space requirements of $\mathcal{O}\left(x^{\frac{1}{3}}\right)$. The timing data Galway provides as Table 5.3 are for sieves of width 10^9 at heights up to 10^{17} (i.e. all 64 bit). The results seem to suggest that the implied constants are quite large and even allowing for the improvements in hardware since 2004, it seemed unlikely that this algorithm would be competitive. However, we have not attempted to implement the dissected Atkin-Bernstein sieve (or the Eratosthenes variant Galway also describes) at large height and it may yet prove workable in practice.

6.4.2 Computing $\phi(p)$

We compute $\phi(p)$ using the following lemma.

Lemma 6.4.2. *Let $0 < \xi < p_0$ and $p \in [p_0 - \xi, p_0 + \xi]$. Then for some $\eta \in [p_0 - \xi, p_0 + \xi]$ we have*

$$\left|\phi(p) - \sum_{n=0}^{m} \frac{\phi^{(n)}(p_0)(p-p_0)^n}{n!}\right| = \left|\frac{\phi^{(m+1)}(\eta)\xi^{m+1}}{(m+1)!}\right|.$$

Proof. Lagrange's form for the error term in Taylor's theorem. □

Simply computing $\phi(p)$ for each the 2.1×10^{13} primes enumerated by the sieve (at $x = 10^{23}$) using the Taylor approximation coded in a multiple precision interval arithmetic package would be prohibitively slow.

To circumvent this, we divide the interval into sub-intervals of width 2ξ and for each sub-interval, centred at t_0, we compute in (fast) integer arithmetic

- $\sum_p 1$
- $\sum_p (p - t_0)$
- $\sum_p (p - t_0)^2$

Note that if we chose the length of sub-interval to be $\leq 2^{32}$, the first two sums can be achieved entirely in 64 bit arithmetic. The last sum requires 32 bit multiplications into 64 bit results which are then summed. Thus the only non 64 bit operation is this summation, but multiple precision integer addition is still relatively fast. Also note that including a $(p-t_0)^3$ term would force us to use non 64 bit multiplication, which we eschew.

These three sums now provide a basis for computing the sum of $\phi(p)$ in multiple precision interval arithmetic.

Even though we wish to avoid using the cubic term of the Taylor approximation, the following lemma obtains some of the improvement in accuracy, effectively for free.

Lemma 6.4.3. *If we approximate the real cubic $y = a_3x^3$ on the interval $x \in [-w, w]$ where $w > 0$ with the line $y = ax$ with $a = \frac{3a_3w^2}{4}$, then the magnitude of the error over the interval is $\leq \frac{|a_3|w^3}{4}$. What is more, in terms of minimising the worst case error, this line is the best choice of any quadratic.*

Proof. We refer to Figure 6.3. Without loss of generality, take $a_3 > 0$ and since both a_3x^3 and ax are odd, we consider only the interval $x \in [0, w]$. The error E_1 is simply $a_3w^3 - aw$ and E_2 is at its maximum where the slopes of the line and the cubic are equal. This happens at $x = \sqrt{\frac{a}{3a_3}}$ so $E_2 = \sqrt{\frac{a^3}{3a_3}} - \sqrt{\frac{a^3}{27a_3}}$. The worst case error follows from setting $E_1 = E_2$ and solving for a.

The maximum error occurs 4 times at $x \in \{\pm w, \pm\sqrt{\frac{a}{3a_3}}\}$. This means that any curve which improves on the line must be below the line at $x \in \{-w, \sqrt{\frac{a}{3a_3}}\}$ and above it at $x \in \{w, -\sqrt{\frac{a}{3a_3}}\}$. Thus, such a curve would have to cross the line at least 3 times which is not possible for a quadratic. □

Thus we can use a quadratic approximation to the cubic Taylor approximation but with a worst case error falling in between the two. We now simply set ξ small enough to achieve the necessary accuracy. At $x = 10^{23}$ we find that $\xi = 2^{21}$ suffices.

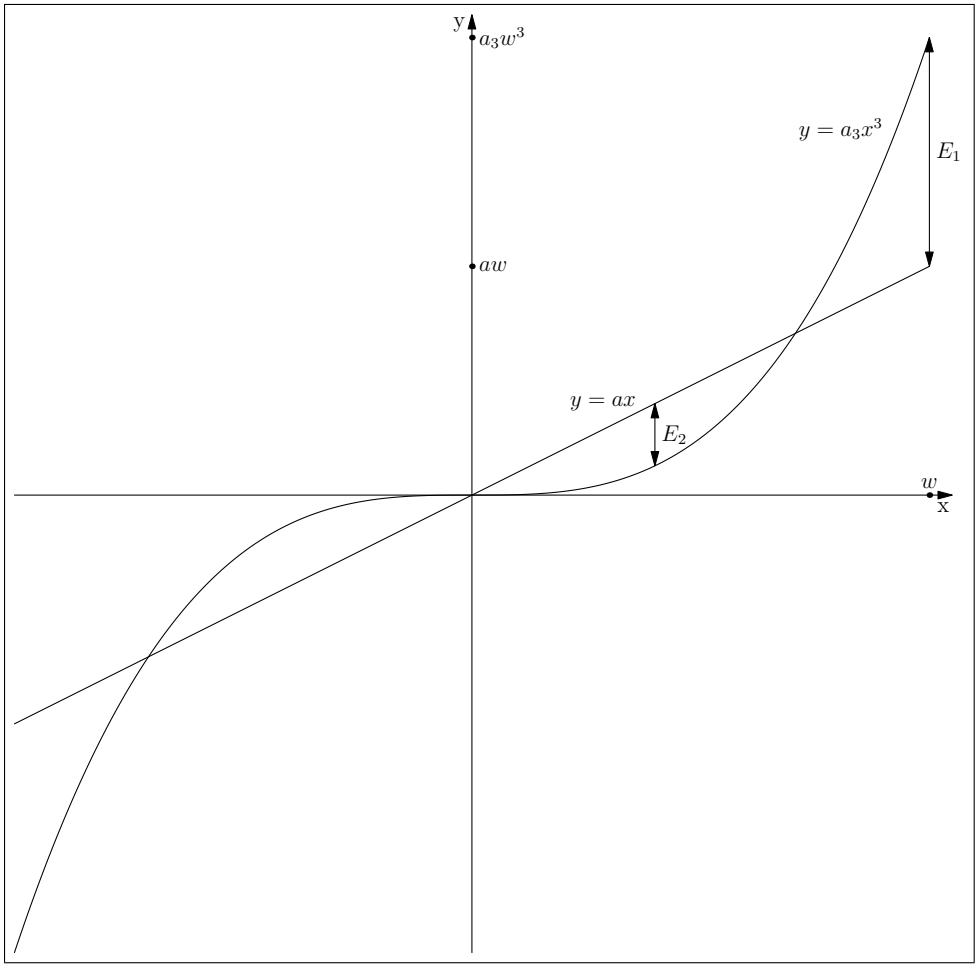

Figure 6.3: Approximating a Cubic with a Line (Lemma 6.4.3)

6.5 Implementation and Results

We set out to confirm the largest unconditionally known value of $\pi(x)$, that for $x = 10^{23}$.

6.5.1 Isolating the Zeros

Using Turing's method with a region of width 42 above and below the target window, we compute the number of zeros we expect to find. Each sign change in f identifies the location of a zero to within $\frac{21}{4\,096}$ (determined by the choices of section 5.11). We then apply a (non-rigorous) version of Newton-Raphson root finding to identify the zeros more precisely, followed by a rigorous up-sampling step to confirm the position of the zero to an absolute accuracy of $\pm 2^{-102}$.

If two or more zeros are located between sample points, then our zero count will not match that predicted by Turing's method. To locate such (pairs of) zeros, we look at the zeros of f'. If the RH is true, then the local maxima and local minima of $f(t)$ will be positive and negative respectively [11]. Thus, there must be (conditional on RH) at least two zeros between any consecutive maxima or minima. Armed with this, we up-sample using a simple bisection algorithm to drill down and locate the sign changes in f. Importantly, because we confirm the existence of such sign changes rigorously, our method is not conditional on RH.

Below $t = 5\,000$ our algorithm is not effective so the zeros below this height were located using mpmath [41] and rigorously checked to $\pm 2^{-102}$ using a high precision interval arithmetic implementation of Euler-Maclaurin.

6.5.2 Parameters for the $\pi(x)$ algorithm

We allowed for an error of slightly less than $\pm\frac{1}{2}$ allocated equally to the truncation of the sum over the ζ zeros (Lemmas 6.3.15 and 6.3.16) and to the truncation of the prime sieve (Lemma 6.4.1).

Selecting the appropriate value of λ depends on the knowledge of the relative run time of the zero locating routine versus the prime sieve (it turns out that the time taken to compute the necessary sums in both cases is insignificant).

We set the parameter λ to, as far a possible, equate the run time of the zero search and the prime sieve. To compute $\pi\left(10^{23}\right)$ we used

- $\lambda = 6\ 224\ 003\ 264\ 759\ 175 \times 2^{-83}$
- $x\exp(\tau) - x\exp(-\tau) = 1\ 154\ 487\ 209\ 164\ 800$, the width of the prime sieve

and we summed over the 36 037 434 430 zeros of ζ with $0 < \Im\rho < 11\ 155\ 646\ 000$.

Both computations were executed on the University of Bristol cluster [2]. Bluecrystal Phase 2 consists of 416 standard compute nodes, each of which houses two 4 core 2.8 GHz Intel Harpertown processors sharing 8 Gbyte of memory. Depending on load, it is possible for a single user to be allocated up to 512 cores (64 nodes) but in practice we typically see just under half this. In total the computation required about 1 000 days of CPU time and the elapsed time was about 6 weeks. Unfortunately, we could not find any published timing data relating to the successful combinatorial computation of $\pi\left(10^{23}\right)$.

6.5.3 Computing $\Re\sum_{\rho}\widehat{\Phi}(\rho)$

We implemented the Taylor series based method described in section 6.3.2, again using 'C' and MPFI. Starting with N zeros numbered 1 to N up the critical line (and with a fictional zero ρ_0 at $\frac{1}{2}+14i$), we compute $\widehat{\Phi}(\rho_n)-\widehat{\Phi}(\rho_{n-1})$ and form the sums

$$\sum_{n=1}^{N}\left[\widehat{\Phi}(\rho_n) - \widehat{\Phi}(\rho_{n-1})\right] = \widehat{\Phi}(\rho_N) - \widehat{\Phi}\left(\frac{1}{2} + 14i\right)$$

and

$$\sum_{n=1}^{N} n(\widehat{\Phi}(\rho_n) - \widehat{\Phi}(\rho_{n-1})) = N\widehat{\Phi}(\rho_N) - \sum_{n=1}^{N-1} \widehat{\Phi}(\rho_n) - \widehat{\Phi}\left(\frac{1}{2} + 14i\right).$$

These sums, together with estimates for $\widehat{\Phi}\left(\frac{1}{2} + 14i\right) - \widehat{\Phi}\left(\frac{1}{2}\right)$ and $\widehat{\Phi}(1) - \widehat{\Phi}\left(\frac{1}{2}\right)$ mean we can compute all the terms of Theorem 6.3.7.

6.5.4 Results

- $a = \Re(\widehat{\Phi}(1) - \widehat{\Phi}\left(\frac{1}{2}\right))$

  $\in [1\ 925\ 320\ 391\ 601\ 622\ 250\ 242.410, 1\ 925\ 320\ 391\ 601\ 622\ 250\ 242.411]$

- $b = \Re(\widehat{\Phi}\left(\frac{1}{2} + 14i\right) - \widehat{\Phi}\left(\frac{1}{2}\right)) \in [-12\ 410\ 224\ 303.294, -12\ 410\ 224\ 303.293]$

- $c = \Re(\widehat{\Phi}(\rho_N) - \widehat{\Phi}\left(\frac{1}{2} + 14i\right)) \in [-21\ 680\ 976.702, -21\ 680\ 976.701]$

- $d = \Re(N\widehat{\Phi}(\rho_N) - \sum_{n=1}^{N-1} \widehat{\Phi}(\rho_n) - \widehat{\Phi}\left(\frac{1}{2} + 14i\right))$

  $\in [-538\ 287\ 888.668, -538\ 287\ 888.640]$

- $e = \sum_{p \in [xe^{-\tau}, xe^{\tau}]} (\chi_x(p) - \phi(p)) \in [-87\ 064.242, -87\ 064.241]$

- $f = \sum_{\substack{p^n \in [xe^{-\tau}, xe^{\tau}] \\ n>1}} \frac{1}{n} (\chi_x(p^n) - \phi(p^n)) \in [0.099, 0.100]$

- $g = \sum_{n>1} \frac{1}{n} \pi\left(x^{\frac{1}{n}}\right) \in [6\ 216\ 885\ 710.894, 6\ 216\ 885\ 710.895]$

- $k = (2N(T_1) - 3)G\left(\frac{1}{2} + iT_1\right) \in [-0.229, -0.228]$

where c and d are the results of the sum over zeros, e and f are the results of the prime sieve and g was computed using Booker's "Nth Prime Page" [12]. a is the result of the integration of a real function along an interval of the real line where it is strictly increasing so any simple quadrature technique can be made rigorous. b uses the same quadrature algorithm as for c and d. Finally k is computed by integrating from $\frac{1}{2} + iT_1$ to $-1 + iT_1$ and adding an error term.

We now have

Theorem 6.5.1. $\pi\left(10^{23}\right) = 1\ 925\ 320\ 391\ 606\ 803\ 968\ 923$.

Proof. Using the parameters above, the error from truncating the sum over zeros and the prime sieve is $\in [-0.301, 0.301]$. We compute the interval $a - b + 2d - 3c + e + f - g - k - \log(2) + [-0.301, 0.301]$ which gives

$$\pi\left(10^{23}\right) \in [1\ 925\ 320\ 391\ 606\ 803\ 968\ 922.665, 1\ 925\ 320\ 391\ 606\ 803\ 968\ 923.333]$$

This brackets a single integer which is therefore the value of $\pi\left(10^{23}\right)$. □

We are pleased to note that this agrees with the figure published by Thomás Oliveira e Silva [63].

Chapter 7

Computing $L_\chi(s)$ Rigorously

The aim of this section is to describe two algorithms for computing $L_\chi(1/2+it)$ rigorously and efficiently and three applications. The first algorithm, used when the modulus is relatively small, is based on the generic method to compute L-functions described in Booker's paper [13]. The other algorithm is new and exploits Theorem 3.2.6.

7.1 Booker's Algorithm

We specialise Booker's generic algorithm from [13] to Dirichlet L-functions.

For $\eta \in (-1,1)$ and even primitive characters χ define

$$F_e(t,\chi) := \epsilon_\chi q^{\frac{it}{2}} \pi^{-\frac{1/2+it}{2}} \Gamma\left(\frac{1/2+it}{2}\right) \exp\left(\frac{\pi\eta t}{4}\right) L_\chi\left(\frac{1}{2}+it\right) \text{ and}$$

$$\hat{F}_e(x,\chi) := \frac{1}{2\pi} \int_{-\infty}^{\infty} F_e(t,\chi) e^{-ixt}\,\mathrm{d}t.$$

For odd primitive characters χ define

$$F_o(t,\chi) := \epsilon_\chi q^{\frac{it}{2}} \pi^{-\frac{3/2+it}{2}} \Gamma\left(\frac{3/2+it}{2}\right) \exp\left(\frac{\pi\eta t}{4}\right) L_\chi\left(\frac{1}{2}+it\right) \text{ and}$$

$$\hat{F}_o(x,\chi) := \frac{1}{2\pi} \int_{-\infty}^{\infty} F_o(t,\chi) e^{-ixt}\,\mathrm{d}t.$$

We chose the parameter η to control the decay of the gamma factor as t increases. ϵ_χ is the square root of the root number ω_χ with argument in $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$.

We have chosen to adhere to the form of the Fourier transform used in [13] for easier cross-reference.

We now chose $A, B > 0$ with $AB \in 2^{\mathbb{Z}_{>0}}$ and define

$$\widetilde{\hat{F}}_e(n,\chi) := \sum_{k\in\mathbb{Z}} \hat{F}_e\left(\frac{2\pi n}{B} + 2\pi kA, \chi\right)$$

and

$$\widetilde{\hat{F}}_o(n,\chi) := \sum_{k\in\mathbb{Z}} \hat{F}_o\left(\frac{2\pi n}{B} + 2\pi kA, \chi\right).$$

Similarly, define

$$\widetilde{F}_e(m,\chi) := \sum_{k\in\mathbb{Z}} F_e\left(\frac{m}{A} + kB, \chi\right)$$

and

$$\widetilde{F}_o(m,\chi) := \sum_{k\in\mathbb{Z}} F_o\left(\frac{m}{A} + kB, \chi\right).$$

In outline, the method is

1. Compute $\hat{F}_e\left(\frac{2\pi n}{B}\right)$ or $\hat{F}_o\left(\frac{2\pi n}{B}\right)$ for $n = 0 \ldots N-1$.
2. Use these values as an approximation to $\widetilde{\hat{F}}_e(n,\chi)$ or $\widetilde{\hat{F}}_o(n,\chi)$ respectively.
3. Appealing to Theorem 3.2.2, perform a DFT to yield $\widetilde{F}_e(m,\chi)$ or $\widetilde{F}_o(m,\chi)$ respectively.
4. Use $\widetilde{F}_e(m,\chi)$ or $\widetilde{F}_o(m,\chi)$ as an approximation to $F_e\left(\frac{m}{A},\chi\right)$ or $F_o\left(\frac{m}{A},\chi\right)$ respectively.

Continuing in the notation of [13], we have $r = 1$, $m = 0$, $P(s) = 1$, $C = 1$, $\alpha = 0$ and $c' = 0$ with the balance of the parameters as per Table 7.1.

We now make the above outline rigorous.

Table 7.1: Parameters

	Even χ	Odd χ
μ_1	0	1
ν_1	0	$\frac{1}{2}$
μ	$\frac{1}{2}$	$\frac{3}{2}$
c	1	2

7.1.1 Computing $\hat{F}_e(t)$ and $\hat{F}_o(t)$

Lemma 7.1.1. *Let $u(x) := \frac{\pi\eta i}{4} + x$, $\delta = \frac{\pi}{2}(1-|\eta|)$ and $X(x) := \frac{\pi\delta e^{2x-\delta}}{q}$. Then for $M \in \mathbb{Z}_{>0}$ and $X(x)M^2 > 1$ we have*

$$\left|\hat{F}_e(x,\chi) - \frac{2\epsilon_\chi \exp\left(\frac{u(x)}{2}\right)}{q^{\frac{1}{4}}} \sum_{n=1}^{M} \chi(n) \exp\left(-\frac{\pi n^2 \exp(2u(x))}{q}\right)\right| \leq \frac{2\exp\left(\frac{x}{2} - X(x)M^2\right)}{q^{1/4}\delta^{1/2}X(x)M}.$$

Proof. Writing $s = 1/2 + it$ we get

$$\begin{aligned}
\hat{F}_e(x,\chi) =& \frac{\epsilon_\chi}{2\pi i} \int\limits_{\Re(s)=\frac{1}{2}} q^{\frac{s-1/2}{2}} \pi^{-\frac{s}{2}} \Gamma\left(\frac{s}{2}\right) \exp\left(\frac{-\pi\eta i(s-1/2)}{4}\right) \exp(-x(s-1/2)) L_\chi(s)\,\mathrm{d}s \\
=& \frac{\epsilon_\chi}{2\pi i} \int\limits_{\Re(s)=2} q^{\frac{s-1/2}{2}} \pi^{-\frac{s}{2}} \Gamma\left(\frac{s}{2}\right) \exp\left(\frac{-\pi\eta i(s-1/2)}{4}\right) \exp(-x(s-1/2)) L_\chi(s)\,\mathrm{d}s \\
=& \frac{\epsilon_\chi}{q^{\frac{1}{4}}} \frac{1}{2\pi i} \int\limits_{\Re(s)=2} \left(\frac{q}{\pi}\right)^{\frac{s}{2}} \Gamma\left(\frac{s}{2}\right) \exp\left(-\left(\frac{\pi\eta i + 4x}{4}\right)(s-1/2)\right) \sum_{n=1}^{\infty} \chi(n) n^{-s}\,\mathrm{d}s \\
=& \frac{\epsilon_\chi \exp(u(x)/2)}{q^{\frac{1}{4}}} \sum_{n=1}^{\infty} \chi(n) \frac{1}{2\pi i} \int\limits_{\Re(s)=2} \left(\frac{\pi n^2}{q}\right)^{-\frac{s}{2}} \Gamma\left(\frac{s}{2}\right) \exp(2u(x))^{-s/2}\,\mathrm{d}s \\
=& \frac{2\epsilon_\chi \exp\left(\frac{u(x)}{2}\right)}{q^{\frac{1}{4}}} \sum_{n=1}^{\infty} \chi(n) \exp\left(-\frac{\pi n^2 \exp(2u(x))}{q}\right).
\end{aligned}$$

We now apply Lemma 5.4 of [13] to bound the error introduced by truncating the above sum at $n = M$. □

Lemma 7.1.2. *Let $u(x)$, δ and $X(x)$ be as defined in Lemma 7.1.1. Then for $M \in \mathbb{Z}_{>0}$ and $X(x)M^2 > 1$ we have*

$$\left|\hat{F}_o(x,\chi) - \frac{2\epsilon_\chi \exp\left(\frac{3u(x)}{2}\right)}{q^{\frac{3}{4}}} \sum_{n=1}^{M} n\chi(n) \exp\left(-\frac{\pi n^2 \exp(2u(x))}{q}\right)\right| \leq$$

$$\frac{2\exp\left(\frac{3x}{2} - X(x)M^2\right)}{q^{3/4}\delta^{1/2}X(x)}\left(1+\frac{1}{2X(x)M^2}\right)^{\frac{1}{2}}.$$

Proof. The proof follows the same lines as Lemma 7.1.1. □

7.1.2 Approximating $\widetilde{\hat{F}}_e$ and $\widetilde{\hat{F}}_o$ with $\hat{F}_e$ and $\hat{F}_o$

We intend to chose our parameters to allow us to use $\hat{F}_e$ and $\hat{F}_o$ as approximations to $\widetilde{\hat{F}}_e$ and $\widetilde{\hat{F}}_o$ respectively. We therefore need to bound the error introduced and we start with a lemma.

Lemma 7.1.3. *For $t \in \mathbb{R}$ we have*

$$\left|L_\chi\left(\frac{1}{2}+it\right)\right| \leq \zeta\left(\frac{9}{8}\right)\left(\frac{q}{2\pi}\right)^{5/16}\left(\frac{3}{2}+|t|\right)^{5/16}.$$

Proof. We evaluate Rademacher's bound [66]

$$|L_\chi(s)| \leq \zeta(1+\eta)\left(\frac{q|1+s|}{2\pi}\right)^{\frac{1+\eta-\Re(s)}{2}}$$

with $\eta = 1/8$ and $s = 1/2 + it$. □

We can now proceed to the necessary bounds.

Lemma 7.1.4. *Let $A \geq \frac{1}{2\pi}$, $B > 0$, $w_1 = \frac{2\pi n}{B} + 2\pi A$, $w_2 = -\frac{2\pi n}{B} + 2\pi A$, with $X(x)$ and δ as defined in Lemma 7.1.1 and $X(w_1), X(w_2) > 1$. Then*

$$\left|\widetilde{\hat{F}}_e(n,\chi) - \hat{F}_e\left(\frac{2\pi n}{B},\chi\right)\right| \leq$$

$$\frac{4\left(\exp\left(\frac{w_1}{2} - X(w_1)\right)\left(1+\frac{1}{2X(w_1)}\right) + \exp\left(\frac{w_2}{2} - X(w_2)\right)\left(1+\frac{1}{2X(w_2)}\right)\right)}{q^{1/4}\delta^{1/2}(1-e^{-\pi A})}$$

and

$$\left|\widetilde{\hat{F}}_o(n,\chi) - \hat{F}_o\left(\frac{2\pi n}{B},\chi\right)\right| \leq$$

$$\frac{4\left(\exp\left(\frac{3w_1}{2} - X(w_1)\right)\left(1+\frac{1}{2X(w_1)}\right)^{\frac{3}{2}} + \exp\left(\frac{3w_2}{2} - X(w_2)\right)\left(1+\frac{1}{2X(w_2)}\right)^{\frac{3}{2}}\right)}{q^{3/4}\delta^{1/2}(1-e^{-\pi A})}.$$

Proof. We apply Lemma 5.6 of [13] with $x = \frac{2\pi n}{B} \pm 2\pi A$. □

Lemma 7.1.5. *Given $t \in \mathbb{R}$ and $B > 0$, we define*

$$E_e(t) := \zeta\left(\frac{9}{8}\right)\pi^{-\frac{1}{4}}\left|\Gamma\left(\frac{1}{4}+\frac{it}{2}\right)\right| e^{\frac{\pi}{4}\eta t}\left(\frac{q}{2\pi}\left|\frac{3}{2}+t\right|\right)^{\frac{5}{16}},$$

$$\beta_e(t) := \frac{\pi}{4} - \frac{1}{2}\arctan\left(\frac{1}{2|t|}\right) - \frac{4}{\pi^2|t^2-\frac{1}{4}|},$$

$$E_o(t) := \zeta\left(\frac{9}{8}\right)\pi^{-\frac{3}{4}}\left|\Gamma\left(\frac{3}{4}+\frac{it}{2}\right)\right| e^{\frac{\pi}{4}\eta t}\left(\frac{q}{2\pi}\left|\frac{3}{2}+t\right|\right)^{\frac{5}{16}}$$

and

$$\beta_o(t) := \frac{\pi}{4} - \frac{3}{2}\arctan\left(\frac{1}{2|t|}\right) - \frac{4}{\pi^2|t^2-\frac{9}{4}|}.$$

Then for $\beta_{e,o}\left(\frac{m}{A}+B\right) > \frac{\pi}{4}\eta$ and $\beta_{e,o}\left(\frac{m}{A}-B\right) > -\frac{\pi}{4}\eta$ we have

$$\left|\widetilde{F}_e(m,\chi) - F_e\left(\frac{m}{A},\chi\right)\right| \leq$$

$$\frac{E_e\left(\frac{m}{A}+B\right)}{1-\exp(-B(\beta_e(m/A+B)-\frac{\pi}{4}\eta))} + \frac{E_e\left(\frac{m}{A}-B\right)}{1-\exp(-B(\beta_e(m/A-B)+\frac{\pi}{4}\eta))}$$

and

$$\left|\widetilde{F}_o(m,\chi) - F_o\left(\frac{m}{A},\chi\right)\right| \leq$$

$$\frac{E_o\left(\frac{m}{A}+B\right)}{1-\exp(-B(\beta_o(m/A+B)-\frac{\pi}{4}\eta))} + \frac{E_o\left(\frac{m}{A}-B\right)}{1-\exp(-B(\beta_o(m/A-B)+\frac{\pi}{4}\eta))}.$$

Proof. We apply Lemma 5.7 (i) of [13] with $t = \frac{m}{A} + B$ and 5.7 (ii) with $t = \frac{m}{A} - B$, replacing the bound for $L_\chi(s)$ with our Lemma 7.1.3. □

We note here that the condition on $\beta_{e,o}(t)$ will fail when t is small, i.e. when $\frac{m}{A} \approx B$. However, this only happens for m approaching AB, by which point the loss of precision through other factors has rendered these values useless for computational purposes anyway.

7.2 A DFT Based Algorithm for $L_\chi(1/2+it)$

Equation 2.4.1 tells us that we can compute Dirichlet L-functions as a sum over Dirichlet characters of values of the Hurwitz zeta function $\zeta(s,\alpha)$. Theorem 3.2.6 tells us how, given those values of $\zeta(s,\alpha)$, we can compute $L_\chi(s)$ in, on average, $\mathcal{O}(\log q)$ operations. The following section discusses a rigorous but fast method of computing Hurwitz zeta.

7.2.1 Computing $\zeta(s,\alpha)$

We need to be able to compute $\zeta(1/2+it, a/q)$ for $1 \le a < q$ with $(a,q)=1$. We use the following lemma:

Lemma 7.2.1. *For $s \notin \mathbb{Z}_{\le 0}$, $\alpha \in (0,1]$ and $|\delta| < \alpha$*

$$\zeta(s,\alpha+\delta) = \sum_{k=0}^{\infty} \frac{(-\delta)^k \zeta(s+k,\alpha)\prod_{j=0}^{k-1}(s+j)}{k!}.$$

Proof. Starting with $\Re s > 1$ and differentiating term by term we have

$$\zeta^{(k)}(s,\alpha) = \sum_{n=0}^{\infty} (-1)^k s(s+1)(s+2)\dots(s+k-1)(n+\alpha)^{-s-k}$$

and the result follows for $\Re s > 1$ by Taylor's Theorem. The Taylor expansion also gives us the analytic continuation to $\mathbb{C} \setminus \mathbb{Z}_{\le 0}$. □

In practice, it is better to work with

$$\zeta_M(s,\alpha) = \zeta(s,\alpha) - \sum_{n=0}^{M} (n+\alpha)^{-s}$$

for some $M \in \mathbb{Z}_{>0}$ and to recover $\zeta(s,\alpha)$ by adding back the missing terms.

To be able to rigorously bound the error in truncating the series definition and the Taylor approximation, we use the following lemmas.

Lemma 7.2.2. *For $\alpha \in (0,1]$, $\Re(s) > 1$ and $M \ge 2$*

$$|\zeta_M(s,\alpha)| \le \frac{(M+\alpha-1)^{1-\Re(s)}}{\Re(s)-1}.$$

Proof. Integral test. □

Lemma 7.2.3. *If we use the first N terms of the Taylor approximation to $\zeta_M(s,\alpha+\delta)$, then the absolute error is bounded by*

$$\frac{(N+1)\,|s(s+1)\dots(s+N-1)\zeta_M(s+N,\alpha)|\,\delta^N}{N!(N+1-(|s|+N)\delta)}$$

and the approximation is valid for

$$\frac{(|s|+N)\delta}{N+1} < 1.$$

Proof. The first term dropped is

$$\frac{s(s+1)\dots(s+N-1)\zeta_M(s+N,\alpha)\delta^N}{N!}$$

and the result follows by considering the geometric sequence with this first term and with common ratio

$$\frac{(|s|+N)\delta}{N+1}.$$

□

7.3 Up-sampling

The output from both algorithms is a lattice of values of $\Lambda_\chi(t)$ (defined in section 2.4). We use the results of section 2.6 to rigorously up-sample.

For $t_0 \in \mathbb{R}$ and $H > 0$ define $W : \mathbb{R} \to \mathbb{R}$ by

$$W(t,\chi) := \Lambda_\chi(t) \exp\left(\frac{-(t-t_0)^2}{2H^2}\right)$$

so $W(t_0,\chi) = \Lambda_\chi(t)$.

We aim to estimate $W(t_0,\chi)$ from our samples using theorems 2.6.1 (Whittaker-Shannon) and 2.6.2. The following lemmas provide the necessary rigorous bounds.

Lemma 7.3.1. *For $a_\chi \in \{0,1\}$*

$$\left|\Gamma\left(\frac{\frac{1}{2}+it+a_\chi}{2}\right)\right| e^{\frac{\pi t}{4}} \leq \max\left(2^{1/4}\sqrt{\pi}\left(\frac{3}{2}+\max(t,0)\right)^{\frac{1}{4}}\exp\left(\frac{1}{6}\right), \sqrt{2\pi}\exp\left(\frac{\pi}{8}+\frac{1}{4}\right)\right).$$

Proof. We use Stirling's approximation separately for $a_\chi = 0$ and $a_\chi = 1$. □

Lemma 7.3.2. *Define I_χ by*

$$I_\chi := 4\int_B^\infty \left|\int_{-\infty}^{\infty} W(t,\chi)\,\mathrm{e}(-xt)\,\mathrm{d}t\right| \mathrm{d}x.$$

Then, writing M in place of $\frac{5}{2} - a_\chi$ we have

$$I_\chi \leq \frac{2\left(\frac{q}{\pi}\right)^{\frac{M}{2}} \zeta(M+1/2) \exp\left(\frac{M^2}{2h^2} - 2\pi BM\right) P(t_0, h)}{M\pi}$$

where

$$P(t_0, h) = \int_{-\infty}^{\infty} \left|\Gamma\left(\frac{3+it}{2}\right)\right| \exp\left(\frac{\pi t}{4}\right) \exp\left(-\frac{(t-t_0)^2}{2h^2}\right) \mathrm{d}t.$$

Proof. Writing $s = 1/2 + it$ we get

$$I_\chi \leq 4 \int_B^\infty \left| \int_{\Re(s)=1/2} \left(\frac{q}{\pi}\right)^{\frac{s-1/2}{2}} \Gamma\left(\frac{s+a_\chi}{2}\right) \exp\left(\frac{\pi i(1/2-s)}{4}\right) L_\chi(s) \right.$$
$$\left. \exp(2\pi(1/2-s)x) \exp\left(\frac{-(i(1/2-s)-t_0)^2}{2h^2}\right) \mathrm{d}s \right| \mathrm{d}x.$$

We now shift the contour of integration to the right so that $\Re(s) = \sigma = 3 - a_\chi$ and write $s = M + 1/2 + it$ to get

$$I_\chi \leq 4 \int_B^\infty \int_{-\infty}^\infty \left| \left(\frac{q}{\pi}\right)^{\frac{M}{2}} \Gamma\left(\frac{3+it}{2}\right) \exp\left(\frac{\pi t}{4}\right) \zeta(M+1/2) \right.$$
$$\left. \exp(-2\pi Mx) \exp\left(\frac{M^2 - (t-t_0)^2}{2h^2}\right) \right| \mathrm{d}t\, \mathrm{d}x.$$

Integrating with respect to t gives us

$$I_\chi \leq 4\left(\frac{q}{\pi}\right)^{\frac{M}{2}} \zeta(M+1/2) \exp\left(\frac{M^2}{2h^2}\right) P(t_0, h) \int_B^\infty \exp(-2\pi Mx)\, \mathrm{d}x$$

and the result follows after integrating with respect to x. □

Lemma 7.3.3. *Let $t_0 \geq 0$. Then*

$$P(t_0, h) \leq h\pi\left(t_0 + \frac{h}{\sqrt{2\pi}} + 1 + \frac{1}{2\sqrt{2}}\right).$$

Proof. We have

$$\begin{aligned}
P(t_0,h) \leq & \int_0^\infty \left|\Gamma\left(\frac{3+it}{2}\right)\right| \exp\left(\frac{\pi t}{4}\right) \exp\left(\frac{-(t-t_0)^2}{2h^2}\right) \mathrm{d}t \\
& + \int_{-\infty}^0 \left|\Gamma\left(\frac{3+it}{2}\right)\right| \exp\left(\frac{\pi t}{4}\right) \exp\left(\frac{-(t-t_0)^2}{2h^2}\right) \mathrm{d}t \\
\leq & \int_0^\infty \frac{(1+t)}{2} \left|\Gamma\left(\frac{1+it}{2}\right)\right| \exp\left(\frac{\pi t}{4}\right) \exp\left(\frac{-(t-t_0)^2}{2h^2}\right) \mathrm{d}t \\
& \qquad + \Gamma\left(\frac{3}{2}\right) \frac{h\sqrt{2\pi}}{2} \left(1 - \operatorname{erf}\left(\frac{\sqrt{2}t_0}{2}\right)\right) \\
\leq & \int_0^\infty \frac{(1+t)}{2} \sqrt{\frac{\pi}{\cosh(\pi t/2)}} \exp\left(\frac{\pi t}{4}\right) \exp\left(\frac{-(t-t_0)^2}{2h^2}\right) dt \\
& \qquad + \Gamma\left(\frac{3}{2}\right) \frac{h\sqrt{2\pi}}{2} \\
\leq & \int_0^\infty \frac{(1+t)}{2} \sqrt{2\pi} \exp\left(\frac{-(t-t_0)^2}{2h^2}\right) \mathrm{d}t + \frac{h\pi\sqrt{2}}{4} \\
\leq & h\pi\left(\frac{h}{\sqrt{2\pi}} + t_0 + 1\right) + \frac{h\pi\sqrt{2}}{4}.
\end{aligned}$$

□

Lemma 7.3.4. *Let $h, B > 0$, $t_0 = \frac{n_0}{2B}$ for some $n_0 \in \mathbb{Z}_{>0}$ and $N \in \mathbb{Z}_{>0}$. Now define*

$$G(n) := \frac{\left(\frac{3}{2} + t_0 + \frac{N+n}{2B}\right)^{9/16} \exp\left(\frac{-(N+n)^2}{8B^2h^2}\right)}{\pi(N+n)}.$$

Then

$$\begin{aligned}
\sum_{n \geq 2Bt_0+N} & \left(\frac{3}{2} + \frac{n}{2B}\right)^{9/16} \exp\left(\frac{-\left(\frac{n}{2B} - t_0\right)^2}{2h^2}\right) \operatorname{sinc}\left(2B\pi\left(\frac{n}{2B} - t_0\right)\right) \\
& \leq \frac{G(0)}{1 - G(1)/G(0)}.
\end{aligned}$$

Proof. $G(n)$ is at least as large as the corresponding term in the sum and the ratio $G(n+1)/G(n)$ is a decreasing function of n so the result follows as the sum of a geometric series. □

We can now combine Lemmas 7.1.3, 7.3.1 and 7.3.4.

Lemma 7.3.5. *Define*

$$E := \sum_{|n|\geq N} W\left(\frac{n}{2B}\right) \operatorname{sinc}\left(2B\pi\left(\frac{n}{2B} - t_0\right)\right).$$

Then for large enough t_0 we have

$$|E| \leq \sqrt{\pi}\zeta\left(\frac{9}{8}\right) \exp(1/6) 2^{5/4} \left(\frac{q}{2\pi}\right)^{5/16} \frac{G(0)}{1 - G(1)/G(0)}.$$

7.4 Application to Rigorous Verification of the GRH

7.4.1 History and Background

The largest rigorous computation to test the GRH for multiple moduli was described by Rumely in [76]. This confirmed the GRH for moduli $q \leq 13$ to height $T = 10\ 000$ and then various q to height $T = 2\ 500$. The largest modulus tested was $q = 432$.

We set out to test the GRH for all moduli $\leq 100\ 000$ up to height T such that $qT \geq 100\ 000\ 000$. There are 1 847 865 074 primitive characters to consider and approximately 9.2×10^{12} zeros to check. This number of zeros is about a factor of 800 000 beyond Rumely and is of the same order of magnitude as the (non-rigorous) computations of Gourdon [36] with Riemann's zeta. We note, however, that the Rumely went on to isolate those zeros to within 10^{-12} and produced statistics on their location. The number of zeros involved our calculation made such analysis impractical.

7.4.2 Method

To be able to isolate the zeros of $L_\chi(s)$, we consider the zero density expected at $qT = 10^8$ which is $\frac{1}{2\pi}\log\left(\frac{qT}{2\pi\,\mathrm{e}}\right) \approx 2.5$. Empirically, we find that sampling at about 5 times this rate strikes a good balance between the cost of computing

more points versus the expense of tracking down zeros that get missed at any given sample rate. We used a spacing $\frac{5}{64}$ or a sample rate of $A = 12.8$.

7.4.2.1 Implementing the DFT based algorithm

We implemented the DFT based algorithm for $q \in [10\,401, 100\,000]$. We first pre-computed a lattice of values of ζ_1 using 'C' and MPFI. Specifically, we computed a lattice for each t from 0 to 9 680 in steps of $\frac{1}{A}$ where an individual lattice consisted of 4097 rows and 15 columns. The j,k'th entry for a given t was $\zeta_1\left(1/2 + k + it, \frac{j}{4096}\right)$. The 0'th row was initialised to $\zeta(1/2 + k + it)$ and the 4097'th row to $\zeta(1/2 + k + it) - 1$. The middle row was computed using the identity

$$\zeta\left(s, \frac{1}{2}\right) = (2^s - 1)\zeta(s)$$

and then the lattice was filled top-down, bottom-up and middle-out using the Taylor approximation (Lemma 7.2.1) or the series definition depending on the size of $\Re s$.

Once computed, each lattice was saved as double precision intervals. In all, we computed nearly 124 000 such lattices and it was the disk space they consumed that limited the maximum t (and therefore minimum q) to which the DFT based algorithm was applicable.

Next, for each q and for each t we used the lattice file to compute the values $\zeta\left(1/2 + it, \frac{a}{q}\right)$ for $(a, q) = 1$ by Taylor approximation with 15 terms and then computed the values $L_\chi(1/2 + it)$ by multi-dimensional DFT as described in Theorem 3.2.6. For primitive χ we then used $L_\chi(1/2 + it)$ to compute $\Lambda_\chi(t)$.

We note that since $\Lambda_\chi(t)$ is real valued, we would be able to deduce (up to sign) ϵ_χ from $L_\chi(1/2 + it)$ and the Γ and π factors. However, we compute all the ϵ_χ for a given modulus via Theorem 3.2.6. Testing that the intervals representing the imaginary parts of $\Lambda_\chi(t)$ all contains zero then acts as a useful internal check.

7.4.2.2 Implementing Booker's algorithm

We implemented Booker's algorithm described in section 7.1 in 'C' using MPFI and 'C++' using int_complex and applied it to $q \in [3, 10\,400]$.

The first part of the computation is the sum in Lemmas 7.1.1 and 7.1.2. These sums consist of $M = \mathcal{O}((qT)^{1/2})$ terms but by summing these terms into their residue classes modulo q and then using Theorem 3.2.6 we save a factor of $\varphi(q)^{1-\epsilon}$. We compute the sums into residue classes using multiple precision intervals with MPFI, then approximate those results with double precision intervals and use our own double precision interval routines from there on.

For small q, the savings of this technique reduce (vanishing for $q \in \{3, 4\}$) and our algorithm has similar complexity to those based on the approximate functional equation. However, over all q, the average time to isolate a single zero remains essentially bounded.

The main technical challenge for small q is the size of the DFT that we need to compute when passing from $\widetilde{\hat{F}}_{e,o}$ to $\widetilde{F}_{e,o}$. For $q = 3$ we wish to compute $L_\chi(1/2+it)$ to height $t = \frac{10^8}{3}$ and to ensure that $\hat{F}$ has decayed sufficiently, we actually need to go at least 5 times higher still. Thus we have $N \approx 5 \times \frac{10^8}{3} \times A \approx 2^{31}$. Since our double precision complex intervals consist of 32 bytes each, our data is of size 2^{36} bytes or 64 Gbytes. The hardware at our disposal was limited to 8 Gbytes of memory, so after allowing for the space to store the roots of unity we were restricted to FFTs of length 2^{27} so we made use of section 3.2.8 to bridge the gap. We note that we could achieve a factor of two saving because one side of the DFT is real, but we did not exploit this.

7.4.2.3 Implementing Turing's method

Regardless of which algorithm has been used to this point, confirming the GRH now reduces to applying Turing's method as discussed in section 2.5.2, potentially after up-sampling to resolve closely spaced zeros using the bounds of section 7.3.

We initially up-sample by a factor of 8, then (if necessary) by 32, 128 and finally by 512. By this stage the step size is $\frac{5}{32768}$ (≈ 0.00015) and only about 0.0003% of the characters remain. By now, we have several possible issues

- A single zero is unaccounted for, but the sign of $\Lambda_\chi(0)$ is indeterminate (i.e. an interval straddling zero). Computing $\Lambda_\chi(0)$ in high precision with Euler-Maclaurin allows us to resolve the true sign and thus locate the missing zero.

- Turing's method produces an upper limit on the number of zeros to find which is non-integral or too large because of pairs of zeros missed in the region above t_0 used to calculate the integral over $\widetilde{N}_{t_0}$. Locating these pairs using high precision resolves the problem.

- Pairs of zeros are missing, but there are regions where, at double precision, $\Lambda_\chi(t)$ is positive (resp. negative), then indeterminate, then positive again (negative). High precision examination of the indeterminate stretch yields a sign change indicating a pair of zeros.

- Pairs of zeros are missing, but there are regions where, at double precision, $\Lambda_\chi(t)$ is positive (resp. negative), indeterminate for many points, then negative (resp. positive). Rather than indicating a single zero, the indeterminate region is hiding three sign changes which are resolved using high precision.

7.4.2.4 Results

Theorem 7.4.1. *The GRH holds for Dirichlet L-functions of primitive character modulus $q \leq 100\,000$ to height T such that $qT = 100\,000\,000$.*

Corollary 7.4.2. *There are no positive real zeros of L_χ for any primitive χ of modulus $q \leq 100\,000$.*

Proof. The existence of a real zero of $L_\chi(s)$ at $s \neq \frac{1}{2}$ would have created a mismatch between Turing's estimate and the number of zeros found. There

were, however, 8 cases where the double precision interval computed for $\Lambda_\chi\left(\frac{1}{2}\right)$ included 0. These cases were all eliminated using higher precision computations. □

7.5 Distribution of Central Values

In [44] Katz and Sarnak investigated the relationship between low lying zeros of L-functions and the eigenvalues of matrices from various ensembles of Random Matrix Theory (RMT). Conrey and Farmer [24] extended this to the mean values of families of L-functions at their central point and Keating and Snaith [46], [45] derived exact expressions for the moments of the characteristic polynomials of matrices with respect to these ensembles at the corresponding point ($\theta = 0$).

Specifically, the distribution of the values of $\Re \log L_\chi(1/2)$ for primitive χ of a given modulus q is expected to resemble the distribution of the values of $\Re \log Z(U, \theta) = \det(I - U\exp(-i\theta))$ of $N \times N$ unitary matrices from the Circular Unitary Ensemble of RMT, when we equate N with $\log q$.

We define (see Equation (6) of [46])

$$M_N(s) := \sum_{j=1}^{N} \frac{\Gamma(j)\Gamma(j+s)}{\Gamma\left(j+\frac{s}{2}\right)^2}.$$

Then (from Equation (36) of [46]) the probability density function r_N for $\Re \log Z$ is approximately

$$r_N(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} M_N(iy)\exp(-ixy)\,\mathrm{d}y. \tag{7.5.1}$$

We compute (non-rigorously) $\zeta\left(\frac{1}{2}, \frac{a}{q}\right)$ for $q = 1\ 000\ 000\ 007$ (a prime) and $a = 1 \ldots q-1$ by Euler-Maclaurin summation. By appealing to Theorem 3.2.6 and using the (non-rigorous) FFTW package [31] on a machine with large memory, we compute $L_\chi(1/2)$ for all $q-2$ primitive characters with a single length $q-1$ DFT. These two steps take a couple of hours of CPU time. We

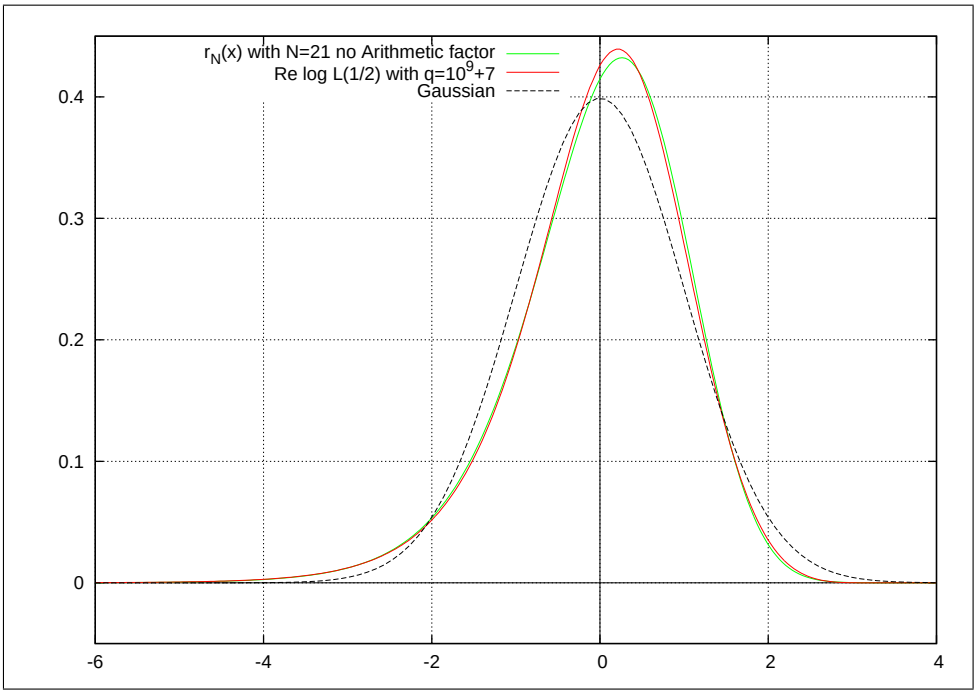

Figure 7.1: $\Re \log L_\chi(1/2)$ vs. RMT Conjecture w/o Arithmetic Factor

then compute $\Re \log L_\chi(1/2)$ and place the values into 500 equal width buckets, then normalise to mean 0 and unit variance.

We now compute a similar number of values for equation 7.5.1 by numerical integration (we used PARI [5] truncating the integral at ± 10), normalised again to 0 mean and unit variance.

We plot both curves, with a normalised Gaussian for reference, as Figure 7.1.

We note that while the two curves are similar, they are discernibly different. In an attempt to at least partially explain this difference, and referring to [46] again, we introduce the arithmetic factor a for ζ as

$$a(\lambda) := \prod_p \left[(1-p^{-1})^{\lambda^2} \left(\sum_{m=0}^{\infty} \left(\frac{\Gamma(\lambda+m)}{m!\Gamma(\lambda)} \right)^2 p^{-m} \right) \right].$$

For the central value of Dirichlet L-functions, this term is modified by a product over primes dividing the modulus (Conjecture 1 of [17]) but for our

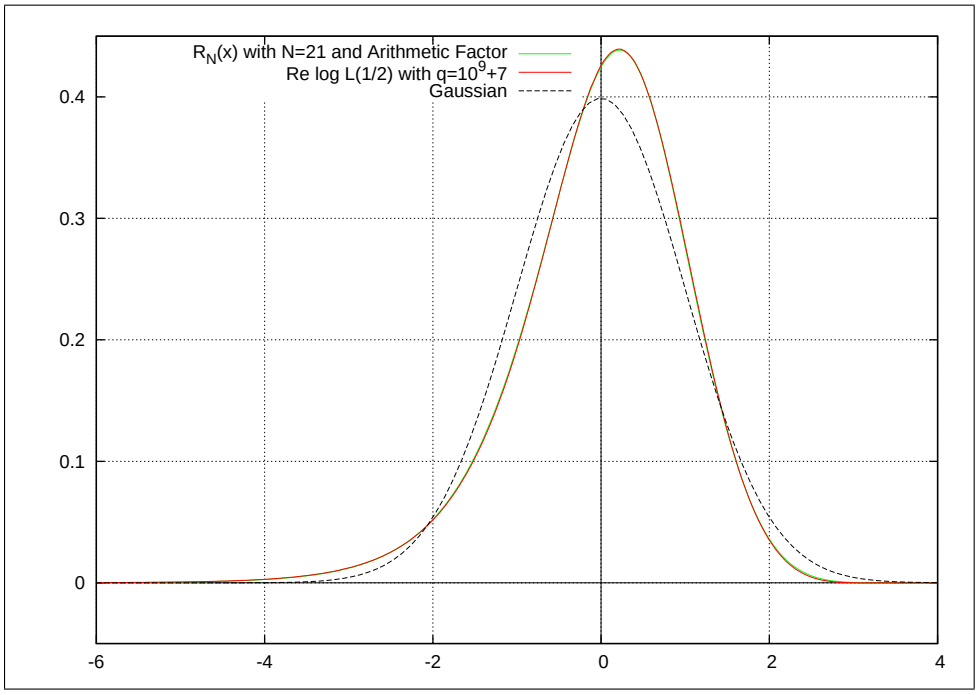

Figure 7.2: $\Re \log L_\chi(1/2)$ vs. RMT Conjecture with Arithmetic Factor

large, prime modulus, this factor will be negligible. Thus we would like to plot

$$R_N(x) := \int_{-\infty}^{\infty} M_N(iy)a(iy/2)\exp(-iyx)\,\mathrm{d}y.$$

In a personal communication, Booker has provided a polynomial approximation to a which allows us to estimate the integral for R_N numerically. The resulting plot is shown as Figure 7.2.

The distribution from computed values of $L_\chi(1/2)$ and that predicted by RMT once the arithmetic factor is included are a significantly better match than before the arithmetic factor was taken into account. It would be interesting to investigate whether this remains true for different q, in particular for q divisible by small primes.

7.6 Non-vanishing of $L_\chi(1/2)$

Chowla conjectured in [20] that $L_\chi(1/2) \neq 0$ for quadratic characters. This has been tested numerically and Watkins [87] shows there are no real positive zeros

for real odd characters of modulus $q \leq 3 \times 10^8$ while Chua [21] established the same for real even characters of modulus $q \leq 2 \times 10^5$. More recently, Omar [64] established the non-vanishing of all real characters with $q \leq 10^{10}$. Our result establishes the same for all primitive characters, real and complex, of modulus $q \leq 2 \times 10^6$.

Theorem 7.6.1. *$L_\chi(1/2) \neq 0$ for all primitive χ of modulus $\leq 2\,000\,000$.*

Proof. We ran the DFT algorithm for $q \in [3, 2\ 000\ 000]$ (739 151 526 102 primitive characters) against the single Hurwitz zeta lattice file for $t = 0$, thus computing $\Lambda_\chi(0)$ as a double precision interval. In 438 152 cases the resulting interval contained zero, so we recomputed those points, again in double precision interval arithmetic, but this time using Euler-Maclaurin summation. This resolved the sign in 438 132 cases, leaving just 20 of indeterminate sign. These in turn were all resolved using a multiple precision interval version of Euler-Maclaurin. □

Chapter 8

Areas for Further Research

What follows is a brief survey of some areas we have identified that may warrant continuing research.

- As we have already observed Rubinstein's "lcalc" [74] computes (non-rigorously) single values for generic L-functions using the smoothed approximate functional equation. The input consists of information about the L-function's poles and its functional equation, together with enough co-efficients of its Dirichlet series. Molin's PhD thesis [55] develops rigorous bounds for the application of double exponential integral formulae. In particular, Molin's results would allow us make Rubinstein's method rigorous. It would be interesting to investigate whether a combination of a rigorous implementation of the smoothed (or windowed) approximate functional equation and FFT techniques could be used to provide a generic calculator to be used when many values of an L-function or a family of L-functions are required.

- Lagarias and Odlyzko observe in their analytic $\pi(x)$ paper [49] that "This technique can be generalized to evaluate many other arithmetic functions, including the functions $\pi(x; k, a)$ counting the number of primes $p \equiv a \pmod{k}$ with $p \leq x$, and the function $M(x)$ which is the partial sum of the Möbius function $\mu(n)$ for all $n \leq x$." Any such new algo-

rithms that rely on ζ and its zeros can obviously re-use the data we have already computed, which might make them an attractive candidate for future research.

- The extreme values of $L_\chi(1)$ for families of Dirichlet L-functions are of interest to number theorists. Our DFT based algorithm for computing Dirichlet L-functions can be readily adapted to this task. Since this algorithm depends on the values of $\zeta(s, \alpha)$ and this has a pole at $s = 1$, we would work instead with a lattice of values of $\lim_{s \to 1+} \zeta(s, \alpha) - \zeta(s)$. We believe we could certainly investigate the behaviour of $L_\chi(1)$ to modulus 10^6 or so.

- We would like to extend our investigations into the non-vanishing of $L_\chi(1/2)$ for general primitive characters beyond our current result for modulus $q \leq 2 \times 10^6$. There are two issues with this. The first is the size of the computation (it is $\mathcal{O}(q^2)$ in time) and the other is the loss of precision we suffer as the size of the DFT's increase. Neither appears insurmountable.

- At the time of writing, we are running the analytic prime counting algorithm to compute $\pi\left(10^{24}\right)$. If successful, this will improve on known unconditional results (including ours) by a factor of 10 and will match the conditional result of Buethe et al. announced in [18].

We have argued that using interval arithmetic to help achieve rigorous computation is both desirable and achievable. However, we have also identified areas for research that would help to encourage this philosophy.

- We discussed in section 3.1.3 some of the challenges in going from real to complex intervals. The approach we adopted, whilst expedient for our applications, will mean that our complex interval class suffers from excessive loss of precision. Tackling the challenge of producing a less

wasteful representation which is not computationally burdensome is, in our opinion, worthy of further effort.

- Our interval arithmetic armoury consists of implementations in double precision and multiple precision. The penalty moving from double to multiple is large, even if the extra precision needed is only a few bits. There might be a role for fixed precision interval packages in quad or even higher formats.

- The desire for high performance graphics capability, particularly for the computer games industry, spurred the development of Graphics Processing Units (GPU's) which sit alongside the work-station's CPU and take away the computational load of rendering, shading and 3-D effects. These multiprocessor devices have now been developed to the point where they can be used as a high performance processing engine for other applications (see for example [60]). Recently, hardware with IEEE compliant floating point processors has been released, importantly including the ability to control rounding modes. Limitations include the I/O bottleneck between the GPU and host CPU, and the relatively small amounts of memory available to the GPU. However, we believe that it would be worthwhile to see if these limitations can be overcome in practice to allow us to exploit the 500 Gflops or so of double precision performance available.

Bibliography

[1] M. Abramowitz and I.A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover publications, 1964.

[2] ACRC. BlueCrystal Phase 2 User Guide, 2009.

[3] Tom M. Apostol. *Introduction to Analytic Number Theory*. Undergraduate Texts in Mathematics. Springer, 1976.

[4] A.O.L. Atkin and D.J. Bernstein. Prime sieves using binary quadratic forms. *Math. Comp.*, 73(246):1023–1030, 2004.

[5] C. Batut, K. Belabas, D. Bernardi, H. Cohen, and M. Olivier. User's Guide to PARI-GP, 2000.

[6] Carter Bays and Richard H. Hudson. The segmented sieve of Eratosthenes and primes in arithmetic progressions to 10^{12}. *BIT*, 17(2):121–127, June 1977.

[7] D.J. Bernstein. primegen, 1999. `http://cr.yp.to/primegen.html`.

[8] M.V. Berry and J.P. Keating. A New Asymptotic Representation for $\zeta(\frac{1}{2} + it)$ and Quantum Spectral Determinants. *Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.*, 437(1899):151–173, 1992.

[9] L. Bluestein. A linear filtering approach to the computation of discrete Fourier transform. *IEEE Transactions on Audio and Electroacoustics*, 18(4):451–455, 1970.

[10] J. Bohman. On the number of primes less than a given limit. *BIT*, 12(4):576–577, 1972.

[11] E. Bombieri. *The Millennium Prize Problems*, chapter The Riemann Hypothesis, pages 107–122. AMS and Clay Mathematics Institute, 2006.

[12] Andrew R. Booker. *The Nth Prime Page* within *The Prime Pages*. `http://primes.utm.edu/nthprime/`.

[13] Andrew R. Booker. Artin's conjecture, Turing's method and the Riemann hypothesis. *Experiment. Math.*, 15(4):385–407, 2006.

[14] J.M. Borwein, D.M. Bradley, and R.E. Crandall. Computational strategies for the Riemann zeta function. *J. Comput. Appl. Math.*, 121(1-2):247–296, 2000.

[15] William L. Briggs and Van Emden Henson. *The DFT An Owners Manual for the Discrete Fourier Transform*. SIAM, 1995.

[16] J.L. Brown Jr. On the Error in Reconstructing a Non-Bandlimited Function by Means of the Bandpass Sampling Theorem. *J. Math. Anal. Appl.*, 18(1):75–84, 1967.

[17] H.M. Bui and J.P. Keating. On the mean values of Dirichlet L-functions. *Proc. Lond. Math. Soc.*, 95(2):273–298, 2007.

[18] Chris Caldwell. Prime Pages, 2010. `http://primes.utm.edu/`.

[19] Y.F. Cheng and S.W. Graham. Explicit estimates for the Riemann zeta function. *Rocky Mountain J. Math.*, 34(4):1261–1280, 2004.

[20] S. Chowla. *The Riemann hypothesis and Hilbert's tenth problem*, volume 4 of *Mathematics and its applications*. Blackie & Son, 1965.

[21] K.S. Chua. Real zeros of Dedekind zeta functions of real quadratic fields. *Math. Comp.*, 74(251):1457–1470, 2005.

[22] B. Cipra. How number theory got the best of the Pentium chip. *Science*, 267(5195):175, 1995.

[23] Mark W. Coffey. An efficient algorithm for the Hurwitz zeta and related functions. *J. Comput. Appl. Math.*, 225(2):338–346, 2009.

[24] J.B. Conrey and D.W. Farmer. Mean values of L-functions and symmetry. *Arxiv preprint arXiv:math/9912107*, 1999.

[25] J.B. Conrey and A. Ghosh. On the Selberg class of Dirichlet series: small degrees. *Duke Math. J*, 72(3):673–695, 1993.

[26] H. Davenport. *Multiplicative Number Theory*. Number 74 in Graduate Texts in Mathematics. Springer, third edition, 2000.

[27] J.A. de Reyna. High Precision Computation of Riemann's Zeta Function by the Riemann-Siegel Formula, I. *Math. Comp*, 80(274):995–1009, 2011.

[28] M. Deléglise and J. Rivat. Computing pi(x): the Meissel, Lehmer, Lagarias, Miller, Odlyzko method. *Math. Comp.*, 65(213):235–246, 1996.

[29] P.G.L. Dirichlet. Beweis eines Satzes über die arithmetische Progression. *Bericht Ak. Wiss. Berlin*, pages 108–110, 1837.

[30] H.M. Edwards. *Riemann's Zeta Function*. Pure and applied mathematics. Academic Press Inc., 1974.

[31] M. Frigo and S.G. Johnson. The Design and Implementation of FFTW3. *Proc. IEEE*, 93(2):216–231, 2005.

[32] W. Gabcke. *Neue Herleitung und explizite Restabschätzung der Riemann-Siegel Formel*. Mathematisch-Naturwissenschaftlichen Fakultät der Georg-August-Universität zu Göttingen, 1979.

[33] W.F. Galway. *Analytic Computation of the Prime Counting Function*. PhD thesis, Univerity of Illinois at Urbana-Champaign, 2004.

[34] I. Gargantini and P. Henrici. Circular arithmetic and the determination of polynomial zeros. *Numer. Math.*, 18(4):305–320, 1971.

[35] X. Gourdon. *The pi(x) Project*. `http://numbers.computation.free.fr/Constants/Primes/Pix/pixproject.html`.

[36] X. Gourdon. The 10^{13} First Zeros of the Riemann Zeta Function, and Zeros Computation at Very Large Height. `http://numbers.computation.free.fr/Constants/Miscellaneous/zetazeros1e13-1e24.pdf`.

[37] D.E.G. Hare. Computing the principal branch of log-Gamma. *J. Algorithms*, 25(2):221–236, 1997.

[38] G.A. Hiary. Fast methods to compute the Riemann zeta function. *Arxiv preprint arXiv:0711.5005*, 2007.

[39] IEEE. IEEE Standard for Binary Floating-Point Arithmetic, IEEE Std 754-1985., 1985.

[40] Intel. *Intel® 64 and IA-32 Architectures Software Developer's Manual*, volume 1-3. Intel, March 2009.

[41] Fredrik Johansson et al. *mpmath: a Python library for arbitrary-precision floating-point arithmetic (version 0.14)*, February 2010. `http://code.google.com/p/mpmath/`.

[42] J. Kaczorowski and A. Perelli. On the structure of the Selberg class, I: $0 \leq d \leq 1$. *Acta Math.*, 182(2):207–241, 1999.

[43] W. Kahan. A logarithm too clever by half., 2004. `http://www.cs.berkeley.edu/ wkahan/LOG10HAF.TXT`.

[44] N.M. Katz and P. Sarnak. Zeroes of zeta functions and symmetry. *Bull. Amer. Math. Soc.*, 36(1):1–26, 1999.

[45] J.P. Keating and N.C. Snaith. Random matrix theory and L-functions at s= 1/2. *Comm. Math. Phys.*, 214(1):91–100, 2000.

[46] J.P. Keating and N.C. Snaith. Random matrix theory and $\zeta(1/2 + it)$. *Comm. Math. Phys.*, 214(1):57–89, 2000.

[47] R. Klatte and C. Ullrich. Complex sector arithmetic. *Computing*, 24(2):139–148, 1980.

[48] A.W. Knapp. *Basic real analysis*, volume 10 of *Cornerstones*. Birkhauser, 2005.

[49] J. C. Lagarias and A. M. Odlyzko. Computing $\pi(x)$: an analytic method. *J. Algorithms*, 8(2):173–191, 1987.

[50] J.C. Lagarias, V.S. Miller, and A.M. Odlyzko. Computing $\pi(x)$: the Meissel-Lehmer method. *Math. Comp.*, 44(170):537–560, 1985.

[51] B. Lambov. *Reliable Implementation of Real Number Algorithms: Theory and Practice*, chapter Interval Arithmetic Using SSE-2. Lecture Notes in Computer Science. Springer, 2008.

[52] R. Sherman Lehman. On the distribution of zeros of the Riemann zeta-function. *Proc. London Math. Soc.*, S3-20(2):303–320, 1970.

[53] D.H. Lehmer. On the exact number of primes less than a given limit. *Illinois J. Math.*, 3(3):381–388, 1959.

[54] R. Lohner and J.W. von Gudenberg. Complex interval division with maximum accuracy. In *Proc. 7th IEEE Symp. on Computer Arithmetic (ARITH 7)(Urbana, Illinois, 1985)*, pages 332–336, 1985.

[55] P. Molin. *Intégration numérique et calculs de fonctions L*. PhD thesis, L'Université Bordeaux I, 2010.

[56] R.A. Mollin. *An introduction to cryptography*. CRC Press, 2007.

[57] R.E. Moore. *Error in digital computation*, volume I, pages 61–130. Wiley, 1965.

[58] R.E. Moore. *Interval analysis*, volume 60. Prentice-Hall Englewood Cliffs, New Jersey, 1966.

[59] J.M. Muller. Correctly Rounded Mathematical Library. `http://lipforge.ens-lyon.fr/www/crlibm/`.

[60] NVIDIA. NVIDIA Home Page. `http://www.nvidia.co.uk/page/home.html`.

[61] A.M. Odlyzko. The 10^{20}-th zero of the Riemann zeta function and 175 million of its neighbors. `http://www.dtc.umn.edu/~odlyzko/unpublished/index.html`, 1992.

[62] A.M. Odlyzko and A. Schönhage. Fast algorithms for multiple evaluations of the Riemann zeta function. *Trans. Amer. Math. Soc.*, 309(2):797–809, 1988.

[63] Tomás Oliveira e Silva. Computing $\pi(x)$: the Combinatorial Method. *Revista do DETUA*, 4(6):759–768, March 2006.

[64] S. Omar. Non-vanishing of Dirichlet L-functions at the central point. In *Proceedings of the 8th International Conference on Algorithmic Number Theory*, pages 443–453. Springer-Verlag, 2008.

[65] M.S. Petkovic and L.D. Petkovic. *Complex interval arithmetic and its applications*. Wiley-VCH, 1998.

[66] H. Rademacher. On the Phragmén-Lindelöf theorem and some applications. *Math. Z.*, 72(1):192–204, 1959.

[67] C.M. Rader. Discrete Fourier transforms when the number of data samples is prime. *Proc. IEEE*, 56(6):1107–1108, 1968.

[68] N. Revol and F. Rouillier. A library for arbitrary precision interval arithmetic. In *10th GAMM - IMACS International Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics*, 2002.

[69] N. Revol and F. Rouillier. Motivations for an arbitrary precision interval arithmetic and the MPFI library. *Reliab. Comput.*, 11(4):275–290, 2005.

[70] B. Riemann. Ueber die Anzahl der Primzahlen unter einer gegebenen Grösse. *Monatsberichte der Berliner Akademie*, November, 1859.

[71] H. Riesel and G. Göhl. Some calculations related to Riemann's prime number formula. *Math. Comp.*, 24(112):969–983, 1970.

[72] J. Rokne and P. Lancaster. Complex interval arithmetic. *Commun. ACM*, 14(2):111–112, 1971.

[73] B. Rosser. Explicit bounds for some functions of prime numbers. *Amer. J. Math.*, 63(1):211–232, 1941.

[74] Michael Rubinstein. *Evidence for a Spectral Interpretation of the Zeros of L-Functions*. PhD thesis, Princeton University, 1998.

[75] Michael Rubinstein. Computational methods and experiments in analytic number theory. In *Recent perspectives in random matrix theory and number theory*, number 322 in London Math. Soc. Lecture Note Ser., pages 425–506. Cambridge Univ. Press, Cambridge, 2005.

[76] R. Rumely. Numerical Computations Concerning the ERH. *Math. Comp.*, 61(203):415–440, 1993.

[77] A. Selberg. Old and new conjectures and results about a class of Dirichlet series. In *Proc. Amalfi Conf. Analytic Number Theory*, pages 367–385. Univ. di Salerno, Salerno, 1989.

[78] C.L. Siegel. Contributions to the theory of the Dirichlet L-series and the Epstein zeta-functions. *Ann. of Math.*, 44(2):143–172, 1943.

[79] R. Šleževičienė. An efficient algorithm for computing Dirichlet L-functions. *Integral Transforms and Spec. Funct.*, 15(6):513–522, 2004.

[80] H. Sorensen, D. Jones, M. Heideman, and C. Burrus. Real-valued fast Fourier transform algorithms. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(6):849–863, 1987.

[81] R. Stallman. *Using GCC: the GNU compiler collection reference manual*. Free Software Foundation, 2003.

[82] The Course Team. *Complex Analysis Handbook*. The Open University, 1994.

[83] E.C. Titchmarsh and D.R. Heath-Brown. *The theory of the Riemann zeta-function*. Oxford University Press, USA, 1986.

[84] T. Trudgian. Improvements to Turing's method. *Math. Comp*, 80(276):2259–2279, 2011.

[85] Alan M. Turing. Some calculations of the Riemann zeta-function. *Proc. Lond. Math. Soc.*, 3(3):99–117, 1953.

[86] J.S. Walker. *Fast Fourier Transforms*. CRC press Boca Raton, 1991.

[87] M. Watkins. Real zeros of real odd Dirichlet L-functions. *Math. Comp.*, 73(245):415–424, 2004.

[88] S. Wedeniwski. ZetaGrid–Computations connected with the Verification of the Riemann Hypothesis. In *Foundations of Computational Mathematics Conference, Minnesota, USA*, 2002.

[89] H. Weyl. *The theory of groups and quantum mechanics*. Courier Dover Publications, 1950.