Formally verified computer-assisted mathematical proofs

Assia Mahboubi (Inria – Vrije Universiteit Amsterdam)
Formalized Mathematics
Representing mathematics in a fixed formal language:
  - Define mathematical objects, statements, proofs;
  - Verify the correctness of proofs by a routine process.

And use a **proof assistant** to make this possible in practice.
Gottfried Wilhelm Leibniz (1646 - 1716)
cf the Calculus Ratiocinator in De Arte Combinatoria, 1666.
An Old Dream

Wiliam Stanley Jevons’ Logical Piano (1870)
... among professional mathematicians

- At best unrealistic and boring;
- At worst uninteresting or even sterilizing;
... among professional mathematicians

- At best unrealistic and boring;
- At worst uninteresting or even sterilizing;

Rares but notable exceptions:
Milestones in program verification:

* Compiler (of a subset of) C;
  CompCert (X. Leroy, 2006)
* A micro-kernel;
  sel4 (G. Klein et al., 2009)
“Mathematica is great for cross-checking numerical values, but it’s not unusual to run into bugs, so triple checking is a good habit.” Fredrik Johansson’s blog (2009)
Applications in Computer-Aided Mathematics?

- “Mathematica is great for cross-checking numerical values, but it’s not unusual to run into bugs, so triple checking is a good habit.” Fredrik Johansson’s blog (2009)
Applications in Computer-Aided Mathematics?

- “Mathematica is great for cross-checking numerical values, but it’s not unusual to run into bugs, so triple checking is a good habit.” Fredrik Johansson’s blog (2009)
- Open source helps but of course does not suffice.
Applications in Mathematics?

“It soon become clear that the only real long-term solution to the problems that I encountered is to start using computers in the verification of mathematical reasoning.”

Vladimir Voevodsky (1966 - 2017)
March 2014, talk at IAS, Princeton
Proof Assistants
Building Blocks

Formal Logic

Proof Assistant

Proof Checker

Libraries
Its choice impacts:

* Reliability of foundations;
* Trustability of the proof checker;
* Modularity of the formalized corpus;
* Automation;
* …
Pioneers: AUTOMATH

* Written by Nicolaas Govert de Bruijn (1918 - 2012); started circa 1967, still usable
* Introduced several key ideas still in use; de Bruijn indices, dependent types, etc.
* Meant to represent mathematics on computer. Edmund Landau’s Foundations of Analysis
Pioneers: Mizar

- Initiated by Andrzej Trybulec (1941 - 2013); started circa 1973
- Based on a flavour of ZFC set theory and first order logic; Tarski-Grothendick set theory
- Meant to represent mathematics on computer. Formalized Mathematics journal
Thm, Nqthm, ACL, ACL2

* Written by Robert Stephen Boyer and J Strother Moore; started circa 1972
* First order, quantifier free logic of computable functions; Lisp
* Designed and used for (industrial) program verification; e.g. Oracle’s SPARC processors
* Strongly automated.
Pioneers: Logic of Computable Functions (LCF)

* Written by Robin Milner (1934 -2010);
  circa 1972, after Dana Scott’s 1969 notes
* Introduced several key ideas still used today;
  Higher-order logic, correctness-by-types, tactics, etc.
* Initiated the ML family of typed programming languages;
  OCaml, Standard ML, etc.
* Designed for program verification.
HOL, HOL88, HOL4, Isabelle/HOL, HOL-Light, HOL-Zero, etc.

* An ambient typed (meta)-programming language;
* An abstract type for theorems, with a few axioms;
* A tactic language for building proofs;
* Church’s simple theory of types;
* Applications in program verification, but also mathematics
* Ernst Zermelo: Untersuchungen über die Grundlagen der Mengenlehre, Mathematische Annalen 65 (2): 261–281

* Bertrand Russell: Mathematical Logic as Based on the Theory of Types, Amer. J. Math. 30 (1908), no. 3, 222–262
Coq, Matita, Agda, Lean

- First prototype by Thierry Coquand and Gérard Huet circa 1984
- Constructive dependent type theory inspired by Automath, Martin Löf, Girard
- Designed for the formalization of mathematics Four Colour Theorem (2004), Odd Order Theorem (2012), Perfectoid spaces (2019), etc.
Verified Computational Mathematics
Conjecture: Johannes Kepler (1611)

Proof strategy, using computer: Lázló Fejes Tóth (1953)


[A formal proof of the Kepler conjecture Hales et al. arxiv]
Generate and:

- Check an exhaustive archive of tame finite graphs;
- Solve linear problems;
- Solve non-linear problems.
Generated by linearization of non-linear ones;
- Infeasibility by feasibility of the dual;
- GLPK used as an oracle to find a solution;
- (Possibly modified) solutions checked inside the logic;
- 43,078 programs, 15 hours on a 2.4GHz computer.

Using HOL-Light
Oracles: Checking vs Proving

- Factorization
- Positivity via decomposition in sums of squares
- Primality checking
- etc.
Flyspeck: Non-Linear Problems

- Non-strict inequalities on rectangular domains;
- Algebraic expressions of transcendental functions;
- Taylor interval approximations;
- 23,000 inequalities, 5000 process-hours in Microsoft’s Azure
- Numbers are an equational theory;
- Computation is normalization in the logic;
- It operates on “logical” floats and integer arithmetics.
Wrapping Up Flyspeck

Issues:

- Time;
- Parallelism without proof objects;
- Two proof assistants: HOL-Light and Isabelle/HOL.
A Verified ODE Solver – Lorentz Attractor

- Introduced by Edward Lorentz (1963);
- 14th problem in Steven Smale’s list (1998);
Fabian Immler (2015):

* implemented a verified ODE solver (2015);
* studied:

\[
f(x, y, z) = (11.8x - 0.29(x + y)z, \\
-22.8y + 0.29(x + y)z, \\
-2.67z + (x + y)(2.2x - 1.3y))
\]

\[
\phi(t) = f(\phi(t))
\]

* verified the numerical computations of the attraction property.

[A Verified ODE Solver and the Lorenz Attractor. – F. Immler. JAR 2018, open access]
Isabelle/HOL allows to:

* translate definitions and functions to SML;
* call evaluation of the translated program;
* trust the (translated back) value inside the logic.
In this work:

- Computations are run in SML;
- Floating point numbers are simulated;
- Integer arithmetic is performed using GMP.

Figures:

- Warwick Tucker’s report: 2000 process-hours
- Fabian Immler’s report: 7000 process-hours
Verified Definite Integrals
Approximating Definite Integrals

\[ m \leq \int_a^b f \leq M \]

- Meaning: integrability of \( f \)
- Correctness: compute and validate bounds
Approximating Definite Integrals

\[
\int_0^8 \sin(t + \exp(t))\, dt \quad \int_{\sqrt{\pi}}^2 \ln(t)\, dt \quad \int_0^{+\infty} \cos(t) \frac{\ln(t)}{t^2}\, dt
\]

* Symbolic integration is not enough.
* Bounds may require approximation.
* Bounds may be singular.

**Motivation**

**Rigorous numerical integration**

Asked 7 years, 7 months ago  Active 7 years, 7 months ago  Viewed 1k times

I need to evaluate some (one-variable) integrals that neither SAGE nor Mathematica can do symbolically. As far as I can tell, I have two options:

(a) Use GSL (via SAGE), Maxima or Mathematica to do numerical integration. This is really a non-option, since, if I understand correctly, the "error bound" they give is not really a guarantee.

(b) Cobble together my own programs using the trapezoidal rule, Simpson's rule, etc., and get rigorous error bounds using bounds I have for the second (or fourth, or what have you) derivative of the function I am integrating. This is what I have been doing.

Is there a third option? Is there standard software that does (b) for me?

**Link**
Goldbach’s Conjectures

* Goldbach’s conjecture:
  Every even integer $\geq 4$ can be expressed as the sum of two primes.

* Goldbach’s ternary (or weak) conjecture:
  Every odd integer $\geq 7$ can be expressed as the sum of three primes.
The ternary Goldbach conjecture is true

H. A. Helfgott

(Submitted on 30 Dec 2013 (v1), last revised 17 Jan 2014 (this version, v2))

The ternary Goldbach conjecture, or three-primes problem, asserts that every odd integer \( n \) greater than 5 is the sum of three primes. The present paper proves this conjecture.

Both the ternary Goldbach conjecture and the binary, or strong, Goldbach conjecture had their origin in an exchange of letters between Euler and Goldbach in 1742. We will follow an approach based on the circle method, the large sieve and exponential sums. Some ideas coming from Hardy, Littlewood and Vinogradov are reinterpreted from a modern perspective. While all work here has to be explicit, the focus is on qualitative gains.

The improved estimates on exponential sums are proven in the author's papers on major and minor arcs for Goldbach's problem. One of the highlights of the present paper is an optimized large sieve for primes. Its ideas get reapplied to the circle method to give an improved estimate for the minor-arc integral.
Minor arcs for Goldbach's problem

H.A. Helfgott

(Submitted on 23 May 2012 (v1), last revised 30 Dec 2013 (this version, v4))

The ternary Goldbach conjecture states that every odd number \( n \geq 7 \) is the sum of three primes. The estimation of sums of the form \( \sum_{p \leq x} e(\alpha p) \), where \( \alpha = a/q + O(1/q^2) \), has been a central part of the main approach to the conjecture since (Vinogradov, 1937). Previous work required \( q \) or \( x \) to be too large to make a proof of the conjecture for all \( n \) feasible.

The present paper gives new bounds on minor arcs and the tails of major arcs. This is part of the author's proof of the ternary Goldbach conjecture.

The new bounds are due to several qualitative improvements. In particular, this paper presents a general method for reducing the cost of Vaughan's identity, as well as a way to exploit the tails of minor arcs in the context of the large sieve.

Comments: 79 pages; third version. (A couple of explanatory paragraphs have been added.)

Major arcs for Goldbach's problem

H.A. Helfgott

(Submitted on 13 May 2013 (v1), last revised 14 Apr 2014 (this version, v4))

The ternary Goldbach conjecture states that every odd number \( n \geq 7 \) is the sum of three primes. The estimation of the Fourier series \( \sum_{p \leq x} e(\alpha p) \) and related sums has been central to the study of the problem since Hardy and Littlewood (1923). Here we show how to estimate such Fourier series for \( \alpha \) in the so-called major arcs, i.e., for \( \alpha \) close to a rational of small denominator. This is part of the author's proof of the ternary Goldbach conjecture. In contrast to most previous work on the subject, we will rely on a finite verification of the Generalized Riemann Hypothesis up to a bounded conductor and bounded height, rather than on zero-free regions. We apply a rigorous verification due to D. Platt; the results we obtain are both rigorous and unconditional. The main point of the paper will be the development of estimates on parabolic cylinder functions that make it possible to use smoothing functions based on the Gaussian. The generality of our explicit formulas will allow us to work with a wide variety of such functions.

Numerical Verification of the Ternary Goldbach Conjecture up to 8,875e30

H.A. Helfgott, David J. Platt

(Submitted on 14 May 2013 (v1), last revised 1 Apr 2014 (this version, v2))

We describe a computation that confirms the ternary Goldbach Conjecture up to 8,875,694,145,621,773,516,800,000,000,000,000 (>8.875e30).

Comments: 4 pages
MAJOR ARCS FOR GOLDBACH’S PROBLEM

By Cauchy-Schwarz, this is at most

\[
\sqrt{\frac{1}{2\pi} \int_{-\frac{1}{2}+i\infty}^{\frac{1}{2}-i\infty} \left| \frac{L'(s, \chi)}{L(s, \chi)} \cdot \frac{1}{s} \right|^2 |ds|} \cdot \sqrt{\frac{1}{2\pi} \int_{-\frac{1}{2}+i\infty}^{\frac{1}{2}-i\infty} \left| G_\delta(s) s^2 \right|^2 |ds|}
\]

By (4.12),

\[
\sqrt{\int_{-\frac{1}{2}-i\infty}^{\frac{1}{2}+i\infty} \left| \frac{L'(s, \chi)}{L(s, \chi)} \cdot \frac{1}{s} \right|^2 |ds|} \leq \sqrt{\int_{-\frac{1}{2}-i\infty}^{\frac{1}{2}+i\infty} \left| \frac{\log q}{s} \right|^2 |ds|}
\]

+ \sqrt{\int_{-\infty}^{\infty} \left| \frac{1}{2} \log \left( \tau^2 + \frac{9}{4} \right) + 4.1396 + \log \pi \right|^2 \frac{1}{\frac{1}{4} + \tau^2} d\tau}

\leq \sqrt{2\pi \log q} + \sqrt{226.844},

where we compute the last integral numerically.\(^4\)

\(^4\)By a rigorous integration from \(\tau = -100000\) to \(\tau = 100000\) using VNODE-LP [Ned06], which runs on the PROFIL/BIAS interval arithmetic package [Knu99].
Floating Point Numbers

Bright side:

* Compromise between precision and range
* IEEE standards
* Efficient algorithms
Floating Point Numbers

Bright side:
* Compromise between precision and range
* IEEE standards
* Efficient algorithms

Dark side:
* Compromise between precision and range
* Notoriously difficult to specify/verify
Interval Arithmetic

\[ 
\mathbb{I} = \{ [a; b] \mid a, b \in \mathbb{R} \cup \{ -\infty \} \cup \{ +\infty \} \} 
\]

* Examples:

\[ [0; 0] \quad [3.14; 3.15] \quad [17; +\infty] \quad [-\infty, +\infty] \]

* Operations:

\[ [-1; 2] \times [-3; 1] = [-6; 2] \quad [0; 100] - [0; 100] = [-100; 100] \]

* Computations: floating-point numbers as bounds.
* Reasoning: interval analysis.
\( F : \mathbb{I}^n \rightarrow \mathbb{I} \) is an interval extension of \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) if:

\[
\forall x \in \mathbb{R}^n, \forall x \in \mathbb{I}^n, x \in x \Rightarrow f(x) \in F(x)
\]

An interval function is isotonic if:

\[
(\forall i, y_i \subset x_i) \Rightarrow F(y_1, \ldots, y_n) \subset F(x_1, \ldots, x_n)
\]
Coarse Bounding

\[ a < b \]

\[ m = \inf_{[a;b]}(f) \]

\[ M = \sup_{[a;b]}(f) \]

If \( f \) is integrable on \([a; b]\) then:

\[ (b - a)m \leq \int_{a}^{b} f \leq (b - a)M \]
Riemann-Darboux Integrability

\[ a = x_0 < x_1 < \cdots < x_n = b \quad m = \inf_{[x_i; x_{i+1}]}(f) \quad M = \sup_{[x_i; x_{i+1}]}(f) \]

\[ f \text{ is integrable on } [a; b] \text{ if:} \]

\[ \sum_{i=0}^{n} (x_i - x_{i-1}) m_i \rightarrow \int_{a}^{b} f \leftarrow \sum_{i=0}^{n} (x_i - x_{i-1}) M_i \]
If $f$ is integrable on $[a; b]$, then:

$$\int_a^b f \in (b - a) \cdot [m; M]$$

If $F$ extends $f$ on $[a; b]$ and $a \in a, b \in b$: 
If $f$ is integrable on $[a; b]$, then:

$$\int_{a}^{b} f \in (b - a) \cdot [m; M]$$

If $F$ extends $f$ on $[a; b]$ and $a \in a, b \in b$:

$$\int_{a}^{b} f \in (b - a) \cdot F(hull(b - a))$$
Using the Coq proof assistant, and its:

- Flocq library
  - floating point arithmetic
- Interval library
  - interval arithmetic and analysis
- Coquelicot library
  - real analysis
The Flocq Library is:

- **A generic library:**
  - Multi-radix, multi-precision;
  - Floating- or fixed-point formats;

- **A multi-purpose tool:**
  - Certification of algorithms;
  - Program verification (Gappa);
  - Computation inside Coq.

* Interval arithmetic on:

\[
I_\bot := \{ [a; b] \mid a, b \in F \} \cup \{ I_\bot \}
\]

\[
IR_\bot := \{ [a; b], [a; +\infty[, ] - \infty, a], ] - \infty, +\infty[ \mid a, b \in \mathbb{R} \} \cup \{ \mathbb{R}_\bot \}
\]

* Interval extensions for mathematical functions;
* Specifications with respect to the concepts in Coquelicot;
* The interval formally verified enclosure procedure.

Coquelicot Library

* Basis of real analysis, with a classical flavour;
* Based on axioms for real numbers;
* Conservative extension of the standard library;
* Defines elementary mathematical functions;
* Includes a theory of Riemann integration.

Based on an abstract syntax of (univariate) expressions:

\[ E ::= x | F | \pi | \sqrt{E} | E^k | E + E | E - E | E \times E | E \div E | - E | \|E\| | \cos(E) | \sin(E) | \tan(E) | \atan(E) | \exp(E) | \ln(E) \]
Expressions are equipped with several evaluations schemes:

- $[e]_R : R \rightarrow R$
- $[e]_{I\bot} : I\bot \rightarrow I\bot$

For any abstract $e$, $t \mapsto [e]_{I\bot}(t)$ is an extension of $x \mapsto [e]_R(x)$:

$$\forall e \in \mathcal{E}, \forall i \in I\bot, \forall x \in i, \quad [e]_R(x) \in [e]_{I\bot}(i)$$
Expressions are equipped with several evaluations schemes:

- \([e]_R : R \rightarrow R\)
- \([e]_{I}\downarrow : I\downarrow \rightarrow I\downarrow\)
- \([e]_{R\downarrow} : R\downarrow \rightarrow R\downarrow\)
- \([e]_{R\downarrow} : R\downarrow \rightarrow R\downarrow\)
Expressions are equipped with several evaluations schemes:

* $[e]_R : R \rightarrow R$
* $[e]_{I_\bot} : I_\bot \rightarrow I_\bot$
* $[e]_{R_\bot} : R_\bot \rightarrow R_\bot$
* Value $\bot_R$ models the complement of the definition domain.
Expressions are equipped with several evaluations schemes:

- \([e]_R : R \rightarrow R\)
- \([e]_{I_\bot} : I_\bot \rightarrow I_\bot\)
- \([e]_{R_\bot} : R_\bot \rightarrow R_\bot\)

- **Value** \(\perp_R\) models the complement of the definition domain.
- **Interval** \(\perp_I\) is the only one containing \(\perp_R\).
Extensions And Formally Verified Correctness

Expressions are equipped with several evaluations schemes:

* $[e]_\mathbb{R} : \mathbb{R} \to \mathbb{R}$
* $[e]_{I_\bot} : I_\bot \to I_\bot$
* $[e]_{\mathbb{R}_\bot} : \mathbb{R}_\bot \to \mathbb{R}_\bot$

* Value $\bot_\mathbb{R}$ models the complement of the definition domain.
* Interval $\bot_I$ is the only one containing $\bot_\mathbb{R}$.

* A stronger correctness theorem holds:

$\forall e \in \mathcal{E}, \forall i \in I_\bot, \forall x \in i, \ [e]_{\mathbb{R}_\bot}(x) \in [e]_{I_\bot}(i)$
In fact, for any expression $e \in \mathcal{E}$ and interval $i$:

$$[e]_{I_{\perp}}(i) \neq I_{\perp} \quad \Rightarrow \quad f \text{ is continuous on } i$$

If $\int_{a}^{b}f$ can be approximated, then $f$ is integrable.
Naive Bounds (Are Not Enough)

* Good enough for small problems:

\[ \int_{0}^{1} (t - t^2 + 12) \, dt \geq 0 \text{ at depth 1} \]

* Very bad for many (many) others:

\[ \int_{0}^{8} \sin(t + \exp(t)) \, dt \in [0.340144; 0.354656] \text{ at depth 20} \]
If $f : \mathbb{R} \rightarrow \mathbb{R}$ is approximated by $P \in \mathbb{R}[X]$ on $[a; b]$:

$$\forall x \in [a; b], f(x) - P(x) \in i$$

Then for any indefinite integral $Q$ of $P$:

$$\int_a^b f \in Q(b) - Q(a) + (b - a)i$$
If \( f : \mathbb{R} \rightarrow \mathbb{R} \) is approximated by \( P \in \mathbb{R}[X] \) on \([a; b]\):

\[
\forall x \in [a; b], \quad f(x) - P(x) \in i
\]

Then for any indefinite integral \( Q \) of \( P \):

\[
\int_{a}^{b} f(x) \in Q(b) - Q(a) + (b-a)i
\]

E.g.: Taylor, Chebyshev expansions
A rigorous polynomial approximation of $f : \mathbb{R} \rightarrow \mathbb{R}$ on $I \in I$ is:

- a pair $(P, \Delta)$;
- with $P \in \mathbb{I}[X]$, $\Delta \in \mathbb{I}$;
- such that there exists a polynomial $P \in \mathbb{R}[X]$ enclosed in $P$ for which $f(x) - P(x) \in \Delta$ for all $x \in I$. 

E.g.: Taylor, Chebyshev models
A rigorous polynomial approximation of \( f : \mathbb{R} \rightarrow \mathbb{R} \) on \( I \subseteq \mathbb{R} \) is:

* a pair \((P, \Delta)\);
* with \( P \in \mathbb{P}[X], \Delta \in \mathbb{P} \);
* such that there exists a polynomial \( P \in \mathbb{R}[X] \) enclosed in \( P \) for which \( f(x) - P(x) \in \Delta \) for all \( x \in I \).

E.g.: Taylor, Chebyshev models
Taylor models are implemented in the Interval library. This results in an extra (correct) evaluation scheme:

\[
\begin{align*}
  [e]_R & : \mathbb{R} \to \mathbb{R} \\
  [e]_{I_\perp} & : I_\perp \to I_\perp \\
  [e]_{R_\perp} & : R_\perp \to R_\perp \\
  [e]_{TM} & : I[X]_{\perp} \times I
\end{align*}
\]

Benchmarks: Easy

\[ \int_0^1 \frac{dx}{1 + x^4} = \frac{\pi}{4} \]

<table>
<thead>
<tr>
<th>Error</th>
<th>Time</th>
<th>Width</th>
<th>Deg</th>
<th>Depth</th>
<th>Prec</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-3}$</td>
<td>0.2</td>
<td>$2^{-10}$</td>
<td>5</td>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>0.2</td>
<td>$2^{-20}$</td>
<td>6</td>
<td>2</td>
<td>30</td>
</tr>
<tr>
<td>$10^{-9}$</td>
<td>0.4</td>
<td>$2^{-30}$</td>
<td>7</td>
<td>3</td>
<td>40</td>
</tr>
<tr>
<td>$10^{-12}$</td>
<td>0.7</td>
<td>$2^{-40}$</td>
<td>10</td>
<td>3</td>
<td>50</td>
</tr>
<tr>
<td>$10^{-15}$</td>
<td>1.0</td>
<td>$2^{-50}$</td>
<td>12</td>
<td>3</td>
<td>60</td>
</tr>
<tr>
<td>$10^{-18}$</td>
<td>1.7</td>
<td>$2^{-60}$</td>
<td>15</td>
<td>3</td>
<td>70</td>
</tr>
</tbody>
</table>
Benchmark: Harder

\[ \int_0^8 \sin(t + \exp(t)) \, dt \simeq 0.3474 \]

<table>
<thead>
<tr>
<th>Error</th>
<th>Time</th>
<th>Width</th>
<th>Deg</th>
<th>Depth</th>
<th>Prec</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-1}$</td>
<td>66.8</td>
<td>$2^{-3}$</td>
<td>6</td>
<td>12</td>
<td>30</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>102.9</td>
<td>$2^{-6}$</td>
<td>5</td>
<td>13</td>
<td>30</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>143.5</td>
<td>$2^{-10}$</td>
<td>6</td>
<td>13</td>
<td>30</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>171.7</td>
<td>$2^{-13}$</td>
<td>7</td>
<td>13</td>
<td>30</td>
</tr>
</tbody>
</table>

[Validated Numerics: A Short Introduction to Rigorous Computations - W. Tucker — suggested by S. Rump]
\[ \int_0^8 \sin(t + \exp(t))\,dt \approx 0.3474 \]

* Octave’s quad, quadcc, and quadgk: off values without any warning;
* Octave’s quadv: off value with a warning;
* Octave’s quadl: 9 seconds to return a correct answer.
* INTLAB’s verifyquad: 1.7 seconds, correct answer;
Benchmark: Entertaining

\[ \int_0^1 (x^4 + 10x^3 + 19x^2 - 6x - 6e^x) \, dx \simeq 11.14731055005714 \]

<table>
<thead>
<tr>
<th>Error</th>
<th>Time</th>
<th>Width</th>
<th>Degree</th>
<th>Depth</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^{-3}</td>
<td>0.6</td>
<td>2^{-10}</td>
<td>5</td>
<td>8</td>
<td>30</td>
</tr>
<tr>
<td>10^{-6}</td>
<td>0.9</td>
<td>2^{-20}</td>
<td>7</td>
<td>13</td>
<td>40</td>
</tr>
<tr>
<td>10^{-9}</td>
<td>1.3</td>
<td>2^{-30}</td>
<td>9</td>
<td>18</td>
<td>50</td>
</tr>
<tr>
<td>10^{-12}</td>
<td>1.9</td>
<td>2^{-40}</td>
<td>11</td>
<td>22</td>
<td>60</td>
</tr>
<tr>
<td>10^{-15}</td>
<td>2.6</td>
<td>2^{-50}</td>
<td>13</td>
<td>28</td>
<td>70</td>
</tr>
</tbody>
</table>
Benchmark: Entertaining

\[ \int_0^1 \left| x^4 + 10x^3 + 19x^2 - 6x - 6e^x \right| dx \approx 11.14731055005714 \]

- Octave’s quad/quadgk: only 10/9 correct digits;
- INTLAB verifyquad: false answer without warning;
- VNODE-LP: cannot be used because of the absolute value.
Benchmark: Improper

\[
\int_{1}^{+\infty} \frac{e^{-x}}{\sqrt{x}} \, dx = \sqrt{\pi} \cdot \text{erfc}(1)
\]

<table>
<thead>
<tr>
<th>Error</th>
<th>Time</th>
<th>Width</th>
<th>Deg</th>
<th>Depth</th>
<th>Prec</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-3}$</td>
<td>0.4</td>
<td>$2^{-10}$</td>
<td>7</td>
<td>3</td>
<td>30</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>1.2</td>
<td>$2^{-20}$</td>
<td>7</td>
<td>5</td>
<td>30</td>
</tr>
<tr>
<td>$10^{-9}$</td>
<td>4.2</td>
<td>$2^{-30}$</td>
<td>9</td>
<td>7</td>
<td>40</td>
</tr>
<tr>
<td>$10^{-12}$</td>
<td>7.7</td>
<td>$2^{-40}$</td>
<td>13</td>
<td>7</td>
<td>50</td>
</tr>
<tr>
<td>$10^{-15}$</td>
<td>17.1</td>
<td>$2^{-50}$</td>
<td>13</td>
<td>8</td>
<td>60</td>
</tr>
</tbody>
</table>
Benchmark: Harder Improper

\[ \int_1^{+\infty} \frac{\cos(x) \ln(x)}{x^2} \, dx \approx 0.1595 \]

Maple 18 forfeits after 10 seconds of computation.
By Cauchy-Schwarz, this is at most
\[
\sqrt{\frac{1}{2\pi} \int_{-\frac{1}{2}-i\infty}^{-\frac{1}{2}+i\infty} \left| \frac{L'(s, \chi)}{L(s, \chi)} \cdot \frac{1}{s} \right|^2 |ds|} \cdot \sqrt{\frac{1}{2\pi} \int_{-\frac{1}{2}-i\infty}^{-\frac{1}{2}+i\infty} |G_\delta(s)|^2 |ds|}
\]

By (4.12),
\[
\sqrt{\int_{-\frac{1}{2}-i\infty}^{-\frac{1}{2}+i\infty} \left| \frac{L'(s, \chi)}{L(s, \chi)} \cdot \frac{1}{s} \right|^2 |ds|} \leq \sqrt{\int_{-\frac{1}{2}-i\infty}^{-\frac{1}{2}+i\infty} \left| \log q \cdot \frac{1}{s} \right|^2 |ds|}
\]
\[
+ \sqrt{\int_{-\infty}^{\infty} \left| \frac{1}{2} \log \left( \tau^2 + \frac{9}{4} \right) + 4.1396 + \log \pi \right|^2 \frac{1}{\frac{1}{4} + \tau^2} d\tau}
\]
\[
\leq \sqrt{2\pi \log q + \sqrt{226.844}},
\]

where we compute the last integral numerically.\(^4\)

\(^4\)By a rigorous integration from \( \tau = -100000 \) to \( \tau = 100000 \) using VNODE-LP [Ned06], which runs on the PROFIL/BIAS interval arithmetic package [Kni99].
Benchmark: Back to Goldbach’s Problem

In fact:

\[
\frac{(0.5 \cdot \ln(\tau^2 + 2.25) + 4.1396 + \ln(\pi))^2}{0.25 + \tau^2}
\]

has to be turned into:

\[
1 + \left( \frac{0.5 \cdot \ln(1 + 2.25/\tau^2) + 4.1396 + \ln(\pi)}{\ln(\tau)} \right)^2 \cdot \frac{\ln^2(\tau)}{\tau^2}
\]

to certify its behavior at \(+\infty\).
Benchmark: Back to Goldbach’s Problem

\[ \int_{100000}^{+\infty} \frac{1 + \left( \frac{0.5 \cdot \ln(1+2.25/\tau^2)+4.1396+\ln(\pi)}{\ln(\tau)} \right)^2}{1 + 0.25/\tau^2} \cdot \frac{\ln^2(\tau)}{\tau^2} \approx 3.17742 \cdot 10^{-3} \]

<table>
<thead>
<tr>
<th>Error</th>
<th>Time</th>
<th>Width</th>
<th>Degree</th>
<th>Depth</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^{-3}</td>
<td>1.1</td>
<td>2^{-10}</td>
<td>3</td>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>10^{-4}</td>
<td>1.6</td>
<td>2^{-13}</td>
<td>5</td>
<td>5</td>
<td>30</td>
</tr>
<tr>
<td>10^{-5}</td>
<td>2.8</td>
<td>2^{-16}</td>
<td>7</td>
<td>9</td>
<td>30</td>
</tr>
<tr>
<td>10^{-6}</td>
<td>5.5</td>
<td>2^{-20}</td>
<td>10</td>
<td>12</td>
<td>30</td>
</tr>
<tr>
<td>10^{-7}</td>
<td>9.8</td>
<td>2^{-23}</td>
<td>12</td>
<td>16</td>
<td>30</td>
</tr>
<tr>
<td>10^{-8}</td>
<td>17.6</td>
<td>2^{-26}</td>
<td>15</td>
<td>18</td>
<td>30</td>
</tr>
</tbody>
</table>

Thus:

\[ \int_{-\infty}^{+\infty} \frac{1 + \left( \frac{0.5 \cdot \ln(1+2.25/\tau^2)+4.1396+\ln(\pi)}{\ln(\tau)} \right)^2}{1 + 0.25/\tau^2} \cdot \frac{\ln^2(\tau)}{\tau^2} \in [226.849; 226.850] \]
Benchmark: Back to Goldbach’s Problem

\[
\int_{100000}^{+\infty} \frac{1 + (0.5 \cdot \ln(1+2.25/\tau^2) + 4.1396 + \ln(\pi))^2}{\ln(\tau) \cdot 1 + 0.25/\tau^2} \cdot \frac{\ln^2(\tau)}{\tau^2} \approx 3.17742 \cdot 10^{-3}
\]

<table>
<thead>
<tr>
<th>Error</th>
<th>Time</th>
<th>Width</th>
<th>Degree</th>
<th>Depth</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-3}$</td>
<td>1.1</td>
<td>$2^{-10}$</td>
<td>3</td>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>1.6</td>
<td>$2^{-13}$</td>
<td>5</td>
<td>5</td>
<td>30</td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>2.8</td>
<td>$2^{-16}$</td>
<td>7</td>
<td>9</td>
<td>30</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>5.5</td>
<td>$2^{-20}$</td>
<td>10</td>
<td>12</td>
<td>30</td>
</tr>
<tr>
<td>$10^{-7}$</td>
<td>9.8</td>
<td>$2^{-23}$</td>
<td>12</td>
<td>16</td>
<td>30</td>
</tr>
<tr>
<td>$10^{-8}$</td>
<td>17.6</td>
<td>$2^{-26}$</td>
<td>15</td>
<td>18</td>
<td>30</td>
</tr>
</tbody>
</table>

Thus:

\[
\int_{-\infty}^{+\infty} \frac{1 + (0.5 \cdot \ln(1+2.25/\tau^2) + 4.1396 + \ln(\pi))^2}{\ln(\tau) \cdot 1 + 0.25/\tau^2} \cdot \frac{\ln^2(\tau)}{\tau^2} \in [226.849; 226.850]
\]

The upper bound 226.844 given in the preprint is incorrect.
Conclusion/Demo
Current version of Helfgott’s proof relies on Arb/MPFR and does not have this error.

Formally verified rigorous computation is gaining traction.

There are many possible lines of improvements from this stub.