A Case Study in Mechanizing Computer-Aided Proofs: Small Gaps Between Primes

Guillaume Baudart[1] and Assia Mahboubi[2]

[1] Inria, France
Guillaume.Baudart@inria.fr
[2] Inria, France and Vrije Universiteit Amsterdam, Netherlands
Assia.Mahboubi@inria.fr

The goal of this project was to mechanise the computational step involved in Proposition 4.3 and formula (8.15) of James Maynard's *Small gaps between primes* [6, 5]. This computer proof validates the lower bound $M_{105} > 4$, where M_k is defined in Proposition 4.2 as the supremum of a certain expression $E(F)$ over the Riemann-integrable functions $F : [0,1]^k \to \mathbb{R}$. The paper reduces the problem to that of bounding the eigenvalues of a certain positive definite matrix. This last step is conducted by a computer program, and the original article refers to an ancillary Mathematica notebook available with the preprint version of the paper [5].

We produced a mechanised counterpart to this notebook. The replacement consists of a FLINT [7] oracle producing untrusted certificates and a Rocq proof based on the MathComp library [8, 4] whose trusted base is the Rocq kernel plus the standard `Uint63` primitive-integer axioms [2]. All proof scripts, lemma statements, infrastructure code, and the certificate emitter were written by Claude under human supervision, with the model driving Rocq through `rocq-mcp` for interactive proof inspection. We used Claude Code [1] with the Opus 4.6 and Opus 4.7 models, together with our `rocq-mcp` [3] tools, which expose Rocq's interactive proof state to the language model. The full development is available at https://github.com/LLM4Rocq/small-prime-gap.

1 The Mechanised Theorem

The main theorem conjoins the spectral bound with the closed-form match of the input matrices, based on Lemmas (8.1) and (8.2). This last step consists in checking that the computation-friendly version of each matrix agrees with its deduction-friendly version, which in turn essentially consists in changing the datastructures respectively used for integers and for matrices.

Note that Lemmas (8.1) and (8.2) are not mechanized, but rather taken as definitions for the coefficients of matrices M_1 (`M1`) and M_2 (`M2`), which in turn justify that of matrix A (`A_rat`):

```
Theorem maynard_M105_certified :
  (forall i j : nat, (i < 42)%nat → (j < 42)%nat →
     M1_spec_ij i j = zrat (mat_get M1_int i j) / zrat D_M1) ∧
  (forall i j : nat, (i < 42)%nat → (j < 42)%nat →
     M2_spec_ij i j = zrat (mat_get M2_int i j) / zrat D_M2) ∧
  exists lambda : realalg,
    eigenvalue (map_mx (ratr : rat → realalg) A_rat) lambda
    ∧ (ratr (4%:Q / 105%:Q) : realalg) < lambda.
```

A single `Print Assumptions maynard_M105_certified` displays the axioms used for establishing both the correctness of the entries-of the two 42×42 matrices and the spectral bound for A: it reports 51 axioms, all standard `PrimInt63`/`Uint63Axioms` primitives, with no project-specific axiom added.

The original Mathematica notebook `Computations.nb` computes a well-chosen eigenvector of the same 42×42 matrix A, snaps it to a small-denominator rational vector v, and evaluates

Table 1: Summary of the experiment

Sessions / date span	8 sessions over $\sim$ 33 days
Assistant / user turns	14,442 / 9,192
Active engagement (model / user)	$\sim$ 100 h (65 h / 35 h)
Total billed tokens	7.09 B
cache read	6.96 B
cache creation	107.8 M
output (generated)	12.7 M
Models used	Opus 4.6 + Opus 4.7
Subagent invocations	224
`rocq-mcp` calls	1,180
Bash calls (total / Rocq-related)	3,703 / 790
Other tool calls (`Edit` / `Read`)	1,226 / 818

the Rayleigh quotient at v using exact rational arithmetic; the resulting value exceeds 4, which provides a rigorous lower bound on the eigenvalues. The current mechanised proof instead computes the characteristic polynomial of the matrix and applies the intermediate value theorem. Because the coefficients are large, computing the characteristic polynomial inside Rocq is delicate: the formal proof compares, modulo enough medium-size primes, the polynomial produced by the FLINT oracle with the one produced by a Rocq implementation of the Faddeev–Le Verrier algorithm, and concludes via the Chinese Remainder Theorem. This strategy was suggested by Claude, which was surprisingly reluctant to auto-formalize a more direct route.

2 Using Claude and Rocq-MCP

The experiment summarized in Table 1 ran across 8 sessions over $\approx$ 33 days, and produced 239 commits. Activity was concentrated in four heavyweight sessions, which account for over 99% of all turns. Proving the main theorem consumed roughly 60% of the total compute budget (7.6 M output tokens, 49 hours of model time). The code-quality audit consumed the remaining 40% (4.9 M output tokens, 17 hours of model time).

Token usage is overwhelmingly dominated by cache reads, as expected for long-running proof sessions where the same blueprint and context are replayed every turn. Generated output totals only $\approx$ 12.7 M tokens. The assistant delegated work to 224 subagents.

Rocq-MCP was used primarily for interactive proof inspection, with 1,180 direct calls to `rocq_check`, `rocq_query`, `rocq_step_multi` and `rocq_start`, plus `rocq_compile_file` as the whole-file verification fallback. A further 790 Rocq invocations went through the shell instead. The cause varied across sessions: in one, the `rocq-mcp` server was simply unavailable, which accounts for a large share of the bash-only `coqc`/`make` calls; in another, `rocq-mcp` usage dropped by roughly 75% following a single context-compaction event, at which point the agent appears to have lost track of the tool descriptions and to have fallen back to driving the compiler directly.

Counting any gap longer than 15 minutes between turns as *user away*, total active engagement across all sessions amounted to roughly 100 hours: about 65 hours of model work versus 35 hours of user steering.

References

[1] Anthropic. Claude Code. https://www.anthropic.com/claude-code.

[2] Michaël Armand, Benjamin Grégoire, Arnaud Spiwack, and Laurent Théry. Extending coq with imperative features and its application to SAT verification. In Matt Kaufmann and Lawrence C. Paulson, editors, *Interactive Theorem Proving, First International Conference, ITP 2010, Edinburgh, UK, July 11-14, 2010. Proceedings*, Lecture Notes in Computer Science, pages 83–98. Springer, 2010.

[3] LLM4Rocq contributors. rocq-mcp: a Model Context Protocol server for Rocq. https://github.com/LLM4Rocq/rocq-mcp.

[4] Assia Mahboubi and Enrico Tassi. *Mathematical Components*. Zenodo, September 2022.

[5] James Maynard. Small gaps between primes. https://arxiv.org/abs/1311.4600, 2013.

[6] James Maynard. Small gaps between primes. *Annals of Mathematics*, 181(1):383–413, 2015.

[7] The FLINT team. *FLINT: Fast Library for Number Theory*, 2026. Version 3.5.0, https://flintlib.org.

[8] The Mathematical Components Team. Mathematical components library. https://math-comp.github.io/.