RAMANUJAN CHALLENGE RESULTS: AI-ASSISTED EVALUATION PIPELINE FOR MULTI-FORMAT PROOFS

Anonymous authors
Paper under double-blind review

ABSTRACT

As AI's mathematical capabilities rapidly improve, meaningful benchmarks are shifting from university-level problems and math olympiads to research-level mathematics. The Ramanujan Challenge is a benchmark comprised of research-level problems in number theory, including unpublished results and open conjectures. It presented concrete formulas for mathematical constants whose correctness can be checked numerically, while their proofs may require non-obvious mathematics. Here, we analyze the Challenge submissions, measuring the relative success of different AI models, Computer Algebra Systems (CAS), and formal tools. We find correct submissions that are conceptually distinct from the official solutions, and a proof of one of the published open conjectures. Evaluating the submissions exposed a second challenge: judging complex proofs reliably at scale. We introduce an AI-assisted evaluation pipeline for natural language proofs, CAS code, and Lean formalizations that assesses semantic and logical correctness. Compared with the conventional one-shot LLM judgment, our pipeline reduces false-positive verdicts on natural language proofs from 31% to 4% and improves judgment accuracy on supporting code from 38% to 57%. This approach provides a methodology for evaluating research-level mathematics at scale, with potential applications across mathematics and AI for science.

1 INTRODUCTION

Benchmarks measuring the mathematical capabilities of AI systems are getting saturated at an accelerating pace. School-level problem sets such as GSM8K and MATH (Cobbe et al., 2021; Hendrycks et al., 2021) were followed by university-level benchmarks, such as U-Math and ASyMOB (Chernyshev et al., 2025; Shalyt et al., 2025), and math olympiad questions (Zheng et al., 2022; He et al., 2024; Gao et al., 2025). The frontier of LLM evaluation is now shifting towards research-level mathematics (Glazer et al., 2025; Phan et al., 2026; Abouzaid et al., 2026; Peyronnet et al., 2026). Meaningful assessment at this level requires a rare combination: problems at the research frontier whose solutions are unpublished to avoid training-data contamination - but are known privately.

In this work, we present the results of the Ramanujan Challenge for AI (Shalyt et al., 2026), a collection of ten research-level numerically verifiable number theory problems. Of these, 8 were proven by the authors but unpublished, and 2 are open conjectures. The collection was released as a benchmark for AI-assisted proving. Our analysis shows that the submissions which were most successful at producing correct natural language proofs used models from the Claude and GPT families, while combining Lean with CAS outperformed reliance on any single tool, suggesting benefits from diversified supporting evidence. Notably, some of the complete proofs were submitted by amateur mathematicians, and one of the open conjectures had multiple correct proofs submitted, showing the potential of advanced AI to lower the barrier of entry to parts of mathematical research.

Obtaining these results required addressing a second challenge: evaluating large quantities of complex mathematical proof candidates. The recent boom in AI-assisted mathematics (Alon et al., 2026; Feng et al., 2026; Ju et al., 2026), while producing some incredible results (OpenAI, 2026; Alpöge, 2026), created an avalanche of mathematical claims – making this problem widespread beyond benchmarks. The Ramanujan Challenge illustrates this scale: participants sent a total of 228 submissions, containing a total of 4802 pages of natural language mathematical proofs and 1,079,380 lines of code across Lean, Python, Mathematica and Sage. Recent mathematical benchmarks ap-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

proach automated validation in different ways: fully relying on a large language model (LLM) as a judge (Zhang et al., 2025), using an LLM to check equivalence against a known ground truth (Peyronnet et al., 2026), or using deterministic code to verify LLM-generated answers (Shalyt et al., 2025). Other benchmarks, such as IMO-GradingBench and μ-Math (Luong et al., 2025; Chernyshev et al., 2025), evaluate LLMs as judges of mathematical proofs and solutions, showing the difficulty of achieving high agreement with expert assessments.

In this paper, we present a novel AI-assisted pipeline, depicted in Figure 1, for the purpose of large scale validation of complex multi-format proofs, which include natural language, CAS code, and Lean formalizations. We assess the Ramanujan Challenge submissions, comparing verdicts across three evaluation methods: our pipeline with and without a human expert in the loop, and a baseline LLM-as-a-judge. Using the evaluation pipeline reduces false positive grading of natural language solutions by a factor of 8, and increases agreement with human evaluation by a factor of 1.5, relative to a baseline of the same LLM without the pipeline.

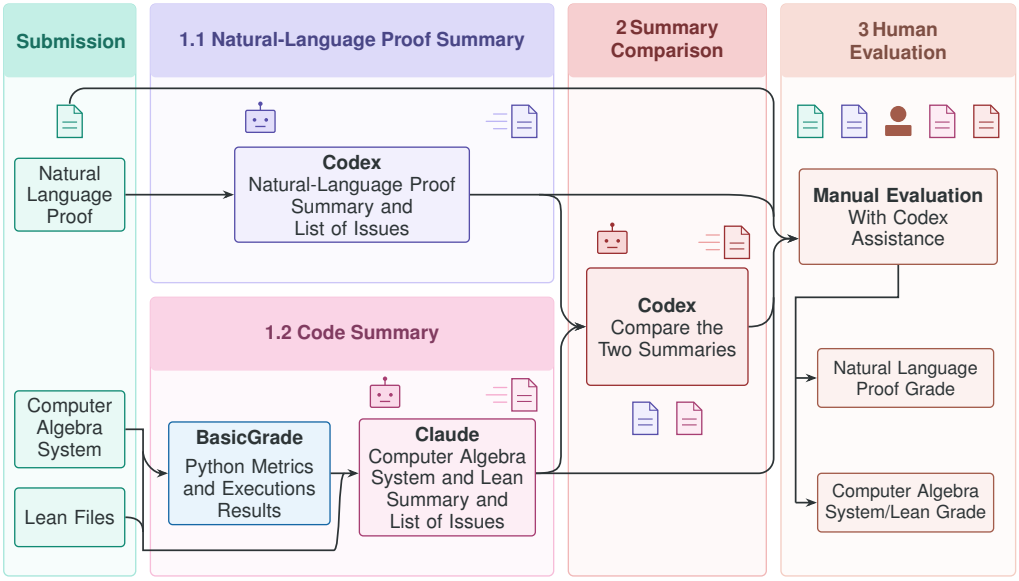

Figure 1: Overview of the evaluation pipeline.

Broadly, as AI systems produce mathematical and scientific results faster than experts can verify them, validation is becoming a central bottleneck (Tao, 2026; Kontorovich, 2025; Codreanu et al., 2026). By combining natural language reasoning with complementary computational certificates, our approach could help scale validation of AI-generated claims across mathematics and science.

2 Related Work and Novelty

Related work. Checking how well LLMs can reason is a relatively new area of interest. Last evaluated in December 2025, the Frontier Math benchmark (Glazer et al., 2025) considered reasoning capabilities of LLMs across multiple domains of research level mathematics. They concluded that the best models of the time were insufficient for advanced research, "revealing a vast gap between AI capabilities and the prowess of the mathematical community". The First Proof (Abouzaid et al., 2026) benchmark focuses on unpublished research problems whose answers are known to experts. These works attest to the high cost of human feedback, and the importance of monitoring the progress of AI as a research assistant. One of the aims of this manuscript is to answer this question for current models.

The other goal is to develop an AI-assisted evaluation pipeline for given solutions. The authors of First Proof stress on the difficulty of manual evaluation when there is a large volume of submissions for complex problems. There are fewer works on the subject of using AI as graders. Using LLMs as teaching assistants for undergraduate discrete math examinations was studied in (Sapkota & Murshed, 2026). The paper suggests that AI can apply harsh point deductions under stricter policy

conditions. LLM as a grader has also been studied in (Xie et al., 2024), where it is used to evaluate responses to six questions in an examination on an Operating Systems Course. Google's DeepMind constructed IMO-GradingBench in (Luong et al., 2025), showing that autograders coupled with reasoning from Gemini seem to correlate well with human evaluation on IMO problems. (Yu et al., 2026) focuses on evaluation of handwritten college mathematics using AI.

Novelty. To the best knowledge of the authors, we are the first to construct an AI-assisted evaluation pipeline as well as test it in the context of large scale evaluations of research-level computational mathematics questions. Our evaluation pipeline, when used by an LLM agent, reduces false positive grading of natural language solutions by a factor of 8, and increases agreement with human evaluation by a factor of 1.5. This demonstrates an improvement in the direction of automated AI verification of natural language complex mathematical proofs.

We further contribute to the analysis of CAS as means of proof or verification for such mathematical questions, an unexplored area of research. We analyze in particular the combined use of CAS and Lean and see that it outperforms the use of either one in isolation, opening a door to further research on the interconnection of CAS and formal theorem assistants in AI mathematics.

3 The Trust Spectrum

Our proposed paradigm for measuring the validity of solutions to mathematical questions is the "trust spectrum". While theoretically most mathematical proofs can be completely formalized, in practice, trust in a mathematical proof lies on a spectrum: soft verification - such as numerical validation or probabilistic checking; intermediate verification - symbolic validation by a computer algebra system (CAS); and hard verification - machine-checked theorem proving like with Lean.

CAS rely on symbolic representations and computational rules rather than axiomatic representations of mathematical objects. They can include data produced by otherwise verifiable mathematical statements, which can be used in subsequent reasoning and proofs in the common sense of the word (Barendregt & Wiedijk, 2005). Provability, however, is not always formally decidable via CAS.

ITPs such as Lean enforce strict rules for line-by-line verification of proofs. They can produce dependency graphs of the proof on other theorems, going right down to the used definitions and axioms. For this reason, we place formal verification the highest on our trust spectrum. Moreover, for a human to verify a formal proof, one needs to inspect only the definitions, used axioms, and problem statement, significantly reducing verification time.

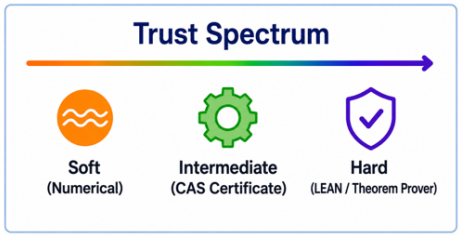

Figure 2: The trust spectrum.

In this paper, one of our aims is to measure how well AI systems can evaluate a given solution, using the trust spectrum as a metric. This view of trust creation in AI results can also generally help decide when to invest substantial time exploring AI outputs, and which parts of a new claim to focus on.

4 The Challenge

The Ramanujan Challenge focuses on explicit formulae involving important mathematical constants such as π, e, $\zeta(s)$ etc. The form they take are of recurrences, continued fractions, series, and integrals. Most of these formulae were discovered by exploring *trajectories* arising in Conservative Matrix Fields, using the `ramanujantools` package (Ramanujan Machine Group, 2025; Elimelech et al., 2024; Weinbaum et al., 2025; Raz et al., 2025).

The Ramanujan Challenge, published on arXiv (Shalyt et al., 2026) and on the Ramanujan Machine website (The Ramanujan Machine, 2026), ran from July to August 2026. It comprised of 8 questions and 2 conjectures, listed in Appendix A and Appendix B respectively. The solutions to these questions were (to the knowledge of the authors) previously unpublished.

The challenge asked for solutions comprising of the following data, in descending order of importance: formal proof, CAS-based derivations and natural language proofs. The reason for this order of importance is discussed in §3.

At the time of posting the Challenge, the prevalent LLMs could not solve any of the problems (with one-shot prompts). The challenge received submissions from people within and outside the mathematics community, many of which did succeed in solving some of the problems.

5 METHODOLOGY

In this section, we discuss the construction of our AI-assisted pipeline, described in Figure 1. The final goal of this pipeline is to reduce human time and effort invested into largescale grading of proofs that require multiple formats.

5.1 EVALUATION PIPELINE

For a submission, the pipeline provides a human reviewer with AI-generated summaries for each file in the submission, along with an AI-generated review connecting all of them.

The major components of the pipeline can be broken down into:
Step 1: Basic Grade Files. Every Python script is executed and given a a "basic grade" using our custom script. This is passed as input in the next step. These files comment on the successful compilation of the given code, as well as summarize the script's structure (comment density, imports, cyclomatic complexity and a keyword-based symbolic/numeric classification) and its observed behavior under a fixed runtime environment (exit status, execution time and standard output).

Step 2: AI Summary. For every submission, two AI summary files are generated, one for the natural language proof and one for the code.

Natural language summary. A GPT 5.6-Sol agent with high reasoning effort reads each PDF file in the submission folder and produces a summary file. The file contains a summary of the proof, a list of issues, and (code) files mentioned in the proof.

CAS Summary. A Claude Opus-5 agent with high reasoning effort reads the CAS/Lean files in each submission folder as well as the basic grade files when present. It produces a file containing a summary of the code, a list of issues, and a list of files that pertain to the verification of the proof.

Note that only relevant files are presented to the agents, not the entire submission folder. This is done to avoid contamination of ideas between the natural language and CAS/Lean segments of the proof. This is done to have a more thorough check of semantic accuracy.

Step 3: Summary Comparison. A GPT 5.6-Sol agent with medium reasoning effort compares the summary files produced in the previous step. It generates a new file listing the similarities between the natural language proof and the code, a list of conflicts, and a list of common issues derived from the two summary files.

Step 4: Human Evaluation. A human can now evaluate the submission with the aid of the summary file(s) generated above.

5.2 EVALUATION CRITERIA

We detail the evaluation criteria by format. Each submission is assigned a rank for its correctness and completeness. Correct and complete proofs or verification codes have been assessed as Correct. Partial ones that demonstrate a meaningful step towards a proof, those with a derivation gap that is not central to the entirety of the proof, or code verification for only part of the proof, have been all classified as Partial. Others with major issues or extra assumptions for the formal proofs have been classified as having Major Issues.

Natural language proofs. Evaluation of natural language proofs is the most self-explanatory. Correct solutions are those that comprise the mathematical steps necessary to reach the conclusion of the statement at hand, with no arrival at statements without proof, except for steps that follow simply

Table 1: Agreement of the LLM agent with human verdicts without using our pipeline (via one-shot prompting) and with the pipeline.

(a) Agreement between human evaluations and those from Claude Opus 5 across submissions to the challenge questions, for natural language proofs and for CAS/Lean code, with one-shot prompting or with the evaluation pipeline.

	Natural language proof		**CAS / Lean**	
Verdict comparison	Without pipeline	With pipeline	Without pipeline	With pipeline
Agreement	60.0%	60.0%	37.8%	56.8%
Claude stricter than human	0.0%	23.8%	2.7%	35.1%
Human stricter than Claude	40.0%	16.2%	59.5%	8.1%

(b) Submissions the human evaluator graded Partial or Major Issues that Claude's proof verdict nonetheless called Correct, as a share of all 80 graded submissions.

Human verdict	Number of submissions	False positives without pipeline	False positives with pipeline
Partial	19	18	3
Major Issues	23	7	0
Total	42 (52.50%)	25 (31.25%)	3 (3.75%)

from standard methods within the field. This is, as all human verification of natural language proofs, susceptible to human error, as discussed in §3.

CAS. CAS code has been evaluated in the same way as a natural language proof. To be classified as correct, it should reproduce all of the steps necessary for a mathematical proof. Code that verified only part of the statements of the natural language proof, or had a small number of seemingly recoverable errors has been deemed partial, and other code, or code that does not compile has been labeled as having major issues.

Lean. Lean was the only ITP used by the participants of the challenge. Its evaluation is much simpler than that of natural language proofs and CAS code. It requires only a semantic inspection of the definitions relating to the problem statement and object, and a check that the code compiles successfully, in addition that the code does not use statements without formal verification via the use of `sorry` or `axiom` statements.

Lean code that does not compile or where the definitions don't match the problem have been classified as having major issues. Code that does compile and matches a significant partial step towards a proof, or code that imports known theorems in the field without formal proof via the use of `sorry` or `axiom` statements have been classified as partial.

6 DATASET

In this section, we explain the various quantitative aspects of the submissions received as part of the Ramanujan Challenge.

The Ramanujan Challenge received 228 submissions. The evaluation so far relies on submissions received by 2026-07-30 22:00 UTC, which total 139 submissions. Evaluation of the later ones will be added to the analysis in the next few weeks.

119 of the submissions clearly relate to a specific challenge problem, with 97 including solutions to the questions and the other 22 to the conjectures. Of these 119, 24 are older versions of later submissions to the same challenge problems by the same submitter. These are ignored in our analysis so that submissions with very little variation in their solution do not skew the data. We do not refer to these hereon.

Participants of the challenge used CAS and ITPs as listed in Figures 3 and 4, with none using any ITPs other than Lean.

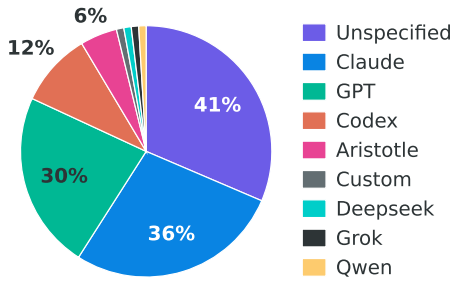

Figure 3: AI models used by the submissions, in percentages out of the total number of submissions. The total percentage exceeds 100% as submissions could include the use of multiple tools.

All submissions answering challenge questions supplied a natural language proof, and all but 6 of those included a formal proof or a CAS-based derivation. The latter are submissions to the challenge questions and can be seen in Figure 4. Most participants reported the AI tools used in their submissions, however, 33 submissions to the challenge questions (Figure 3), and 4 to the conjectures, did not provide this information. We believe that submissions that did not specify the use of AI tools did not avoid their use, but rather only omitted the specification of said tools. As such, we analyze these submissions as part of the data, with specific mention where it is excluded.

3 participants had only submissions that don't relate to a specific challenge problem. 10 participants submitted only 1 solution while 4 who submitted solutions to all of those problems. The conjectures received 11 and 4 solutions respectively, with 3 people who submitted to both of them.

The type of formality of the solutions is detailed in Figure 5, where a solution was classified as having the lowest type of formality appearing in its CAS/Lean code, according to §3

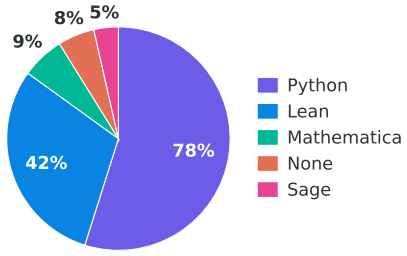

Figure 4: Computer Algebra Systems or Lean used by the submissions, in percentages out of the total number of submissions. The total percentage exceeds 100% as submissions could include the use of multiple tools.

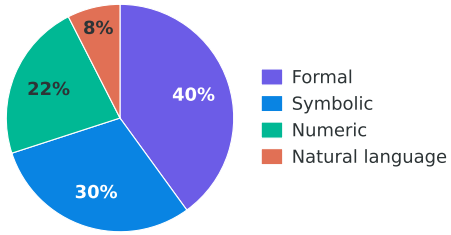

Figure 5: Type of formality of the submissions, in percentages out of the total number of submissions.

7 Results

We now present the results of evaluating the submissions to the Ramanujan Challenge. In §7.1, we detail our primary quantitative findings, including evaluation of the correctness of the solutions, similarity metrics and novelty analysis. §7.2 details our observations regarding the open challenge questions. §7.3 details the successes of our AI-assisted evaluation pipeline.

7.1 Analysis of the Solutions to the Questions

Figure 7 shows the success rates in submissions to the questions, as measured by the human reviewer separately for the natural language proof and for the verification code.

Of the reported AI tools, a combination of Claude and GPT or Codex achieves highest correctness for both natural language proofs and CAS/Lean correctness.

The highest success for natural language proofs was achieved by those that did not include any CAS code, though these were produced by only two different submitters. This is followed by submissions using Python in combination with Lean, which also performed the best as a means of proof verification. A combination of CAS and Lean obtained overall highest correctness across both natural language and code correctness, as seen in Figure 6.

Table 2: Human evaluation of the submissions, with each submission receiving a trinary grade for its natural language proof, CAS and Lean code, when present.

		Natural language proof			Computer Algebra System/Lean			
Question	Number of submissions	Correct	Partial	Major issues	Correct	Partial	Major issues	No code
1	17	8	5	4	6	7	3	1
2	9	5	2	2	0	6	2	1
3	11	8	1	2	2	5	3	1
4	12	3	5	4	1	8	2	1
5	5	1	1	3	0	2	3	0
6	12	8	3	1	2	7	1	2
7	8	3	1	4	2	4	2	0
8	6	2	1	3	0	2	4	0
Total	80	38	19	23	13	41	20	6

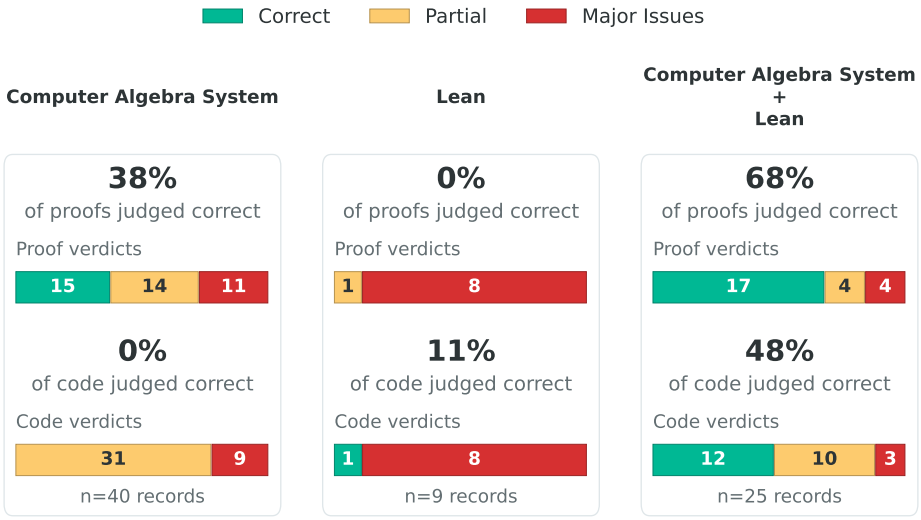

Figure 6: Correctness of natural language proofs and CAS derivation or Lean proofs, according to human evaluation.

7.2 Solutions to the Conjectures

We present here the results regarding the two conjectures of the challenge, which are described in Appendix B. Conjecture 1 received solutions including a correct Lean-formalized proof and multiple other proofs that were deemed likely correct after discussion with the problem author. Among the evaluated submissions, Conjecture 2 received only partial solutions, and it remains open. We discuss these in more detail.

Due to scope and time constraints, the only solutions to Conjecture 1 that were evaluated were those presenting a solution in Lean, and partially those which only gave a natural language proof. Discussion with the author of the problem suggests that multiple solutions are, with varying degrees of mathematical rigor, more-or-less correct. Evaluation of the Lean files for this problem shows one correct and complete formal proof, with two of the other three solutions having major issues and not formalizing the theorem statement, and one providing proof only for partial results.

Evaluation of Conjecture 2 found only partial solutions or those with major issues, with all solutions admitting to only attempt proving partial results. One of the solutions supplies Lean definitions and a formalization of the conjecture, without providing an actual proof for it. The other three solutions all demonstrated promising direction towards their claims, albeit with different issues. They all relied on numerical checks for the verification of their statements. One had certain mathematical

errors in their derivation, and the other supplied dense and hard-to-read texts which prevented their complete and rigorous evaluation.

7.3 Analysis of the Evaluation Pipeline

We analyze the evaluation pipeline as a method of increasing performance of LLM agents in the correct classification of challenge solutions. A Claude Opus 5 agent was used twice, once via one-shot prompting and once with the use of our evaluation pipeline, to classify the parts of the solution as correct, partial or with major issues. These were compared to human verdicts, which were considered the reference labels, with results detailed in Table 1.

We see that the agent was never too strict when using one-shot prompting in the evaluation of natural language proofs. It was too lax on 40% of the submissions, ignoring issues with the solutions. Also, 31.25% of all such proofs had been incorrectly classified as correct. Both of these issues are corrected with the use of the pipeline, with the agent being stricter on 23.8% of solutions and only incorrectly classifying solutions as correct on 3.75% of solutions, less by a factor of 8 than the number without the pipeline.

Analysis of CAS or Lean code shows a stronger improvement. Agreement with the human verdict increases from 37.8% to 56.8% with the use of the pipeline, and the part of solutions for which it was too lax decreased from 59.5% down to 8.1%, by a factor of 7. It did not, however, classify any solutions as correct, which we believe could be improved with different prompts in the use of the pipeline, and further discuss in §9.

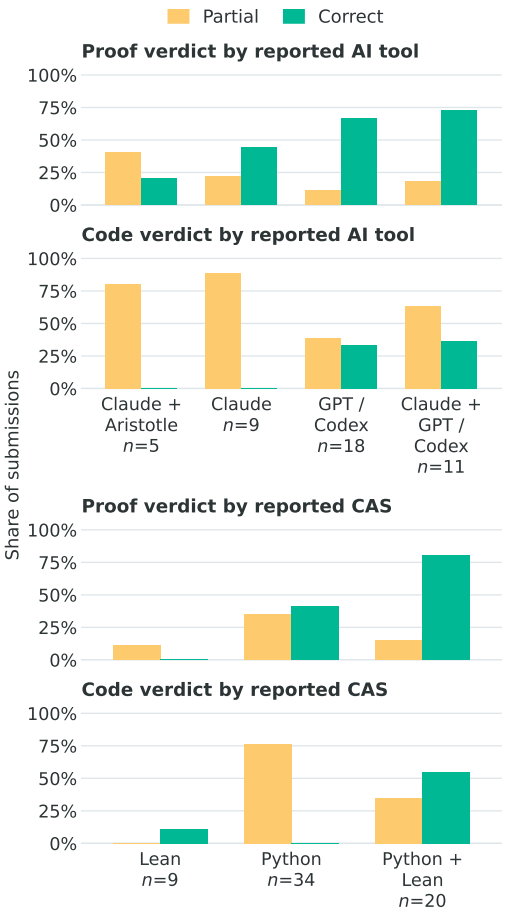

Figure 7: Correct and Partial verdicts for submissions to the challenge questions, by tool combination, for combinations that appeared in at least 5 submissions.

8 Limitations

There is uncertainty in the experiment results with regards to variation between participants and tools used. Challenge participants submitted solutions to as low as 1 and as high as all 10 challenge problems, potentially skewing the dataset. The participants were furthermore self-admitted to the challenge and chose and self-reported their use of tools for the solutions, introducing potential biases.

Evaluation of the submissions also leads to an amount of uncertainty, as they rely on human judgment, which is prone to error. Due to time and monetary constraints, evaluation of each natural language proof and CAS or Lean code was done by a single human evaluator, as described in §5 and without cross-checking. This leaves evaluation prone to human error and potentially skewed by the specific backgrounds of the evaluators.

9 Discussion and Outlook

As AI-assisted mathematics produces results at an increasing rate, reliable and scalable validation is becoming crucial. Our analysis shows that a structured evaluation pipeline can improve judgment

quality even when the underlying LLM is unchanged. We expect this advantage to persist as models improve, particularly if evaluation pipelines evolve in parallel to model improvement.

The Ramanujan Challenge results show that research-level number theory problems are increasingly within reach of AI-assisted solvers. Benchmarks at this level may therefore saturate as well, just as earlier benchmarks have, motivating a continuous effort to construct up-to-date evaluations that track the advancing research frontier. Open conjectures offer a renewable source of such problems, but because they lack known proofs, their use as benchmarks depends on reliable evaluation methods that do not require a priori ground truth to work well, such as the approach presented here.

Our analysis shows that submissions supported by both Lean and CAS code were, on average, more successful than submissions relying on either a single tool or using no computational certificates. One plausible explanation is that the two forms of evidence are complementary: Lean guarantees the logical validity of the statement as formalized, whereas CAS can efficiently check intricate symbolic and numerical computations that are hard to formalize. Together, they can therefore provide stronger evidence than either alone. This suggests a broader role for CAS as a supporting tool for mathematical proofs. CAS certificates are often more readable than full formalizations while remaining machine-checkable, and tighter integration among CAS, proof assistants, and LLMs could turn verified computations into certified proof steps that combine readability with rigor.

More broadly, the setting considered here – a natural language claim accompanied by code-based indirect evidence – extends beyond number theory. Similar situations arise throughout mathematics and increasingly in varied areas of AI for science, where generated claims may be supported by code, simulations, or data analyses. As the volume of such results grows, systems capable of autonomously assessing the validity of a claim and the sufficiency of its supporting evidence may become increasingly crucial.

10 AI USE

Our work relies heavily on the use of LLMs. The components where AI was majorly used include:
The dataset. All or most challenge solutions were generated by AI to some extent. A detailed analysis of this is provided in §6. Since these are external contributions, the authors do not have access to the amount of tokens used for all submissions.
Evaluation of solutions. Codex with GPT-5.6-Sol with high reasoning effort and Claude Opus 5 with high reasoning effort were used heavily in the evaluation evaluation pipeline. Codex with GPT-5.6 was also used as an assistant during manual human evaluation of the solutions in addressing different questions regarding derivation gaps and proof outlines, with answers only inspected by the human reviewer as part of their own evaluation.
Graphs. Code was generated with the help of AI to analyze and create graphs.
Survey. LLMs were used for the purpose of literature survey.
Grammar and phrasing. Feedback regarding correct grammar and phrasing.

The following was the contribution differentiated based on the models that were used:
Claude. We used a total of about 250,000,000 tokens over Opus 4.8, Opus 5, and Fable 5. We worked on a Claude Max subscription costing 100 USD per month for 2 months.
Codex. We used about 436,988,144 tokens, with 84,848,451 used for the natural language proof summary and 5,059,975 used in the summary comparison files. 347,079,718 tokens were used in the interactive sessions during human evaluation of the submissions.
We confirm not having used generative AI tools for any other tasks related to this manuscript, and take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

11 ETHICS STATEMENT

This manuscript and all supporting documents adhere to the ICLR Code of Ethics. We have taken care to avoid any biases or discriminatory outcomes in our research process. We have made sure to maintain the anonymity of the submitters to the Ramanujan Challenge. We are committed to maintaining transparency and integrity throughout the research process.

12 Reproducibility Statement

Anonymized submission results are included in the supplemental materials. For each submission, the listed data includes the tools used in that submission, human evaluation results and claude evaluation results for that submission with and without the pipeline. The order of appearance in the file, within each problem of the challenge, is randomized so that the results and tools used cannot be inferred from the public list of submissions on the challenge website.

References

Mohammed Abouzaid, Andrew J. Blumberg, Martin Hairer, et al. First proof. *arXiv preprint arXiv:2602.05192*, 2026. doi: 10.48550/arXiv.2602.05192. URL https://arxiv.org/abs/2602.05192.

Noga Alon, Thomas F. Bloom, W. T. Gowers, Daniel Litt, Will Sawin, Arul Shankar, Jacob Tsimerman, Victor Wang, and Melanie Matchett Wood. Remarks on the disproof of the unit distance conjecture, 2026. URL https://arxiv.org/abs/2605.20695.

Levent Alpöge. A counterexample to the Jacobian conjecture in dimension three, July 2026. URL https://x.com/__alpoge__/status/2079028340955197566. Announcement, July 19, 2026.

Roger Apéry. Irrationalité de $\zeta(2)$ et $\zeta(3)$. In *Journées Arithmétiques de Luminy*, volume 61 of *Astérisque*, pp. 11–13. Société Mathématique de France, 1979.

Henk (Hendrik) Barendregt and Freek Wiedijk. The challenge of computer mathematics. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 363:2351–73; discussion 2374, 09 2005. doi: 10.1098/rsta.2005.1650.

Konstantin Chernyshev, Vitaliy Polshkov, Ekaterina Artemova, Alex Myasnikov, Vlad Stepanov, Alexei Miasnikov, and Sergei Tilga. U-math: A university-level benchmark for evaluating mathematical skills in llms, 2025. URL https://arxiv.org/abs/2412.03205.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Mihai Codreanu, Alex Imas, Juan Mateos-Garcia, Joseph Emmens, Evalyne Muiruri, Arthur Turrell, Julian Jacobs, Atoosa Kasirzadeh, Ana Trisovic, Yiyuan Chen, Tanya Rodchenko, Catherine Pollard, Scott Strand, Daniel Rock, Zanna Iscenko, Fabien Curto Millet, Neil Thompson, and James Manyika. Ai in science: Early insights, 2026. URL https://arxiv.org/abs/2609.28504.

Rotem Elimelech, Ofir David, Carlos De la Cruz Mengual, Rotem Kalisch, Wolfgang Berndt, Michael Shalyt, Mark Silberstein, Yaron Hadad, and Ido Kaminer. Algorithm-assisted discovery of an intrinsic order among mathematical constants. *Proceedings of the National Academy of Sciences*, 121(25):e2321440121, 2024. doi: 10.1073/pnas.2321440121. URL https://www.pnas.org/doi/abs/10.1073/pnas.2321440121.

Tony Feng, Trieu H. Trinh, Garrett Bingham, Dawsen Hwang, Yuri Chervonyi, Junehyuk Jung, Joonkyung Lee, Carlo Pagano, Sang hyun Kim, Federico Pasqualotto, Sergei Gukov, Jonathan N. Lee, Junsu Kim, Kaiying Hou, Golnaz Ghiasi, Yi Tay, YaGuang Li, Chenkai Kuang, Yuan Liu, Hanzhao Lin, Evan Zheran Liu, Nigamaa Nayakanti, Xiaomeng Yang, Heng-Tze Cheng, Demis Hassabis, Koray Kavukcuoglu, Quoc V. Le, and Thang Luong. Towards autonomous mathematics research, 2026. URL https://arxiv.org/abs/2602.10177.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. Omni-MATH: A universal olympiad level mathematic benchmark for large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=yaqPf0KAlN.

Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, Olli Järviniemi, Matthew Barnett, Robert Sandler, Matej Vrzala, Jaime Sevilla, Qiuyu Ren, Elizabeth Pratt, Lionel Levine, Grant Barkley, Natalie Stewart, Bogdan Grechuk, Tetiana Grechuk, Shreepranav Varma Enugandla, and Mark Wildon. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai, 2025. URL https://arxiv.org/abs/2411.04872.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.211. URL https://aclanthology.org/2024.acl-long.211/.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

Haocheng Ju, Guoxiong Gao, Jiedong Jiang, Bin Wu, Zeming Sun, Shurui Liu, Leheng Chen, Yutong Wang, Yuefeng Wang, Zichen Wang, Wanyi He, Peihao Wu, Liang Xiao, Ruochuan Liu, Bryan Dai, and Bin Dong. Automated conjecture resolution with formal verification, 2026. URL https://arxiv.org/abs/2604.03789.

V. T. Khoi. On the integral of $\log x \, \frac{dy}{y} - \log y \, \frac{dx}{x}$ over the A-polynomial curves. *Acta Mathematica Vietnamica*, 33(3):519–528, 2008. URL https://arxiv.org/abs/0811.2725.

Alex Kontorovich. The shape of math to come, 2025. URL https://arxiv.org/abs/2510.15924.

Thang Luong, Dawsen Hwang, Hoang H. Nguyen, Golnaz Ghiasi, Yuri Chervonyi, Insuk Seo, Junsu Kim, Garrett Bingham, Jonathan Lee, Swaroop Mishra, Alex Zhai, Clara Huiyi Hu, Henryk Michalewski, Jimin Kim, Jeonghyun Ahn, Junhwi Bae, Xingyou Song, Trieu H. Trinh, Quoc V. Le, and Junehyuk Jung. Towards robust mathematical reasoning, 2025. URL https://arxiv.org/abs/2511.01846.

OpenAI. Finite time blowup for Navier–Stokes. Preprint, September 2026. URL https://cdn.openai.com/pdf/32d9f210-8b73-45e0-91bc-82a30aef8a9a/navier-stokes.pdf. Accompanied by Lean formalization; not peer reviewed.

Antoine Peyronnet, Fabian Gloeckle, and Amaury Hayat. Lemmabench: A live, research-level benchmark to evaluate llm capabilities in mathematics, 2026. URL https://arxiv.org/abs/2602.24173.

Long Phan, Alice Gatti, Nathaniel Li, Adam Khoja, Ryan Kim, Richard Ren, Jason Hausenloy, Oliver Zhang, et al. A benchmark of expert-level academic questions to assess ai capabilities. *Nature*, 649(8099):1139–1146, January 2026. ISSN 1476-4687. doi: 10.1038/s41586-025-09962-4. URL http://dx.doi.org/10.1038/s41586-025-09962-4.

Ramanujan Machine Group. ramanujantools, 2025. URL https://zenodo.org/doi/10.5281/zenodo.15747099.

Tomer Raz, Michael Shalyt, Elyasheev Leibtag, Rotem Kalisch, Shachar Weinbaum, Yaron Hadad, and Ido Kaminer. From euler to ai: Unifying formulas for mathematical constants. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 123959–124040. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/b394adff21327520268aaf565b63414c-Paper-Conference.pdf.

Aastha Sapkota and M. G. Sarwar Murshed. Llms as teaching assistants for mathematics exam grading: Reliability, and practical usability, 2026. URL https://arxiv.org/abs/2607.01247.

Michael Shalyt, Rotem Elimelech, and Ido Kaminer. Asymob: Algebraic symbolic mathematical operations benchmark, 2025. URL https://arxiv.org/abs/2505.23851.

Michael Shalyt, Rotem Kalisch, Carsten Schneider, Hila Barkan, Elyasheev Leibtag, John Campbell, Shachar Weinbaum, Tali Monderer, Ashvni Narayanan, and Ido Kaminer. The ramanujan challenge for ai, 2026. URL https://arxiv.org/abs/2607.09721.

Terence Tao. Mathematics in the age of ai, 2026. URL https://arxiv.org/abs/2608.16753.

The Ramanujan Machine. Ramanujan challenge. https://ramanujanmachine.com/ramanujan-challenge/, 2026. Accessed: September 22, 2026.

Shachar Weinbaum, Elyasheev Leibtag, Rotem Kalisch, Michael Shalyt, and Ido Kaminer. On conservative matrix fields: Continuous asymptotics and arithmetic, 2025. URL https://arxiv.org/abs/2507.08138.

Wenjing Xie, Juxin Niu, Chun Jason Xue, and Nan Guan. Grade like a human: Rethinking automated assessment with large language models, 2024. URL https://arxiv.org/abs/2405.19694.

Zhiqi Yu, Xingping Liu, Haobin Mao, Mingshuo Liu, Long Chen, Jack Xin, and Yifeng Yu. Evaluating AI grading on real-world handwritten college mathematics: A large-scale study toward a benchmark. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=vcdsMrNGqP.

Jie Zhang, Cezara Petrui, Kristina Nikolic, and Florian Tramer. Realmath: A continuous benchmark for evaluating language models on research-level mathematics. *arXiv preprint arXiv:2505.12575*, 2025. doi: 10.48550/arXiv.2505.12575. URL https://arxiv.org/abs/2505.12575.

Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. minif2f: a cross-system benchmark for formal olympiad-level mathematics. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=9ZPegFuFTFv.

A THE CHALLENGE QUESTIONS

Question 1 (Polynomial continued fraction for π)**.** *Let:*

$$a_n = -220n^3 - 484n^2 - 301n - 42, \qquad b_n = 4n^2(2n+1)^2(5n-4)(5n+6). \tag{1}$$

Prove:

$$a_0 + \cfrac{b_1}{a_1 + \cfrac{b_2}{a_2 + \cfrac{b_3}{a_3 + \ddots}}} = \frac{6}{3-\pi}. \tag{2}$$

Question 2 (Euler's constant γ as an Apéry limit)**.** *For $n \geq 0$, define:*

$$\begin{aligned} 0 = &\left(-8n^3 - 51n^2 - 105n - 68\right) u_n \\ &+ \left(24n^5 + 337n^4 + 1833n^3 + 4818n^2 + 6092n + 2928\right) u_{n-1} \\ &- (n+2)(n+3)\left(24n^5 + 273n^4 + 1150n^3 + 2154n^2 + 1635n + 268\right) u_{n-2} \\ &+ (n+1)(n+2)^4(n+3)\left(8n^3 + 75n^2 + 231n + 232\right) u_{n-3} \end{aligned}$$

Let p_n, q_n be two solutions of the recurrence, defined by the initial values:

$$p_{-3} = 0, \quad p_{-2} = 7, \quad p_{-1} = 179$$
$$q_{-3} = 1, \quad q_{-2} = 12, \quad q_{-1} = 306$$

Prove

$$\lim_{n\to\infty} \frac{p_n}{q_n} = \gamma,$$

where γ is Euler's constant.

Question 3 (The sum $\pi + e$ as an Apéry limit)**.** *For $n \geq 1$, define*

$$\begin{aligned} 0 = &\left(-n^3 + 2n^2 + 7n + 3\right) u_n \\ &+ (n+2)\left(2n^4 + n^3 - 26n^2 - 48n - 19\right) u_{n-1} \\ &+ (n+2)\left(n^6 + 9n^5 + 8n^4 - 87n^3 - 249n^2 - 234n - 68\right) u_{n-2} \\ &+ (n+1)^2(n+2)\left(2n^5 + 3n^4 - 13n^3 - 21n^2 + 4\right) u_{n-3} \\ &- n^3(n+1)^2(n+2)\left(n^3 + n^2 - 8n - 11\right) u_{n-4}. \end{aligned}$$

Let p_n, q_n be two solutions of the recurrence, defined by the initial values:

$$p_{-3} = 1, \quad p_{-2} = 1, \quad p_{-1} = 20, \quad p_0 = 296$$
$$q_{-3} = 1, \quad q_{-2} = 0, \quad q_{-1} = 4, \quad q_0 = 48$$

Prove

$$\lim_{n\to\infty} \frac{p_n}{q_n} = \pi + e.$$

Question 4 (Harmonic numbers, a polylogarithm and zeta values)**.** *Prove:*

$$\sum_{m=0}^{\infty} \sum_{k=0}^{m} \frac{\binom{m}{k}^2 H_k^2}{(m+1)^2 \binom{2m}{m}} =$$

$$20\,\mathrm{Li}_4\left(\frac{1}{2}\right) + \frac{5}{6}\log^4(2) + 10\zeta(2) - \frac{65}{9}\zeta(2)^2 - \log^2(2)(12 + 5\zeta(2)) + \frac{1}{2}\zeta(3) + \log(2)\left(\frac{35}{2}\zeta(3) - 16\right),$$

where $H_0 = 0$ and for $k \geq 1$, H_k is the k-th harmonic number.

Question 5 (Efficient rational approximation of Catalan's constant G)**.** *For $n \geq 0$, let*

$$M(n) = \big(m_{ij}(n)\big)_{1\leq i,j\leq 3},$$

where

$$\begin{aligned}
m_{11}(n) &= (-2n-5)(n+3)^2(136n^4+1424n^3+5548n^2+9551n+6141),\\
m_{12}(n) &= 384n^6+6384n^5+44168n^4+162698n^3+336377n^2+369933n+169011,\\
m_{13}(n) &= -480n^4-4980n^3-19210n^2-32690n-20730,\\
m_{21}(n) &= (n+2)^2(n+3)^2(4n+10)(48n^3+386n^2+1017n+879),\\
m_{22}(n) &= (n+2)^2(-272n^5-3848n^4-21732n^3-61184n^2-85761n-47808),\\
m_{23}(n) &= (n+2)^2(320n^3+2540n^2+6610n+5640),\\
m_{31}(n) &= (-4n-10)(n+2)^2(n+3)^2(32n^4+302n^3+1037n^2+1530n+813),\\
m_{32}(n) &= (n+2)^2(192n^6+2984n^5+19116n^4+64452n^3+120256n^2+117279n+46476),\\
m_{33}(n) &= (n+2)^2(-16n^5-408n^4-2912n^3-8884n^2-12254n-6240).
\end{aligned}$$

For $N \geq 0$, define

$$\mathcal{M}_N = M(0)M(1)\cdots M(N-1).$$

Choose initial conditions

$$A = \begin{pmatrix} 30921 & -32972 & 8240 \\ 33750 & -36000 & 9000 \end{pmatrix}.$$

Write

$$A\mathcal{M}_N = \begin{pmatrix} P_{N,1} & P_{N,2} & P_{N,3} \\ Q_{N,1} & Q_{N,2} & Q_{N,3} \end{pmatrix}.$$

Then, for each $j = 1, 2, 3$, prove:

$$\lim_{N\to\infty} \frac{P_{N,j}}{Q_{N,j}} = G,$$

where

$$G = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)^2}$$

is Catalan's constant.

Question 6 (A series for $\zeta(2)+\zeta(3)$)**.** *Let $(u_n)_{n\geq 1}$ be the sequence defined by*

$$u_1 = -\frac{93}{4480}, \qquad u_2 = -\frac{117}{14000},$$

and, for $n \geq 3$, by the recurrence

$$\begin{aligned}
0 = &-2(n+3)^3(2n+5)(3n+5)u_n\\
&+(n+2)^2(15n^3+85n^2+155n+93)u_{n-1}\\
&-(n+1)^3(n+2)(3n+8)u_{n-2}.
\end{aligned}$$

Prove:

$$\frac{2077}{720} + \sum_{j=1}^{\infty} u_j = \zeta(2)+\zeta(3).$$

Question 7 (Efficient four-term recurrence for $\zeta(2)+\zeta(3)$)**.** *For $n \geq 0$, define*

$A_n = 1024(2n+5)^4(2n+7)^3(2n+9)^3\big(946n^2+6407n+10860\big)$,
$B_n = 128(2n+7)^3(2n+9)^3\big(104060n^6+1745370n^5+12145238n^4+44886481n^3+92943995n^2+102256019n+46709052\big)$,
$C_n = 16(n+3)^4(2n+9)^3\big(3784n^5+57792n^4+351019n^3+1059230n^2+1587211n+944620\big)$,
$D_n = (n+3)^4(n+4)^6\big(946n^2+4515n+5399\big)$.

Let $(p_n)_{n\ge 0}$ and $(q_n)_{n\ge 0}$ be the two solutions of the recurrence

$$u_{n+1} = \frac{B_n}{A_n}\,u_n - \frac{C_{n-1}}{A_{n-1}}\,u_{n-1} + \frac{D_{n-2}}{A_{n-2}}\,u_{n-2}, \qquad n \ge 2,$$

with initial conditions

$$p_0 = -612218384750, \qquad p_1 = -\frac{9525021973931919}{18100}, \qquad p_2 = -\frac{29561828382772029}{65380},$$

and

$$q_0 = -215040420000, \qquad q_1 = -\frac{167282265043404}{905}, \qquad q_2 = -\frac{964185327658080}{6071}.$$

Prove:

$$\lim_{n\to\infty} \frac{p_n}{q_n} = \zeta(2) + \zeta(3).$$

Question 8 (Very fast rational approximation of $\sqrt{10005}/\pi$)**.** *Let*

$$R = 151931373056001, \qquad u = 2n+3, \qquad w = u(3u-2)(3u+2).$$

For $n \ge 0$, let

$$M(n) = \begin{pmatrix} \dfrac{a_1}{w} & \dfrac{a_2}{w} & \dfrac{a_3}{w} & \dfrac{144R(u-1)^2}{w} \\ -u^3 & -3u^2 & -3u & -1 \\ \dfrac{b_1}{144R} & -\dfrac{b_2}{72R} & -\dfrac{b_3}{36R} & -\dfrac{2u+2R-7}{2R} \\ \dfrac{c_1}{288R^2} & \dfrac{c_2}{144R^2} & \dfrac{c_3}{72R^2} & \dfrac{c_4}{4R^2} \end{pmatrix},$$

where

$$\begin{aligned} a_1 &= (144R-99)u^5 - (288R-333)u^4 + (144R-229)u^3 - 114u^2 + 40u + 64, \\ a_2 &= (432R-243)u^4 - (864R-909)u^3 + (432R-868)u^2 - 80u + 272, \\ a_3 &= (432R-153)u^3 - (864R-648)u^2 + (432R-860)u + 360, \\ b_1 &= 9u^4 - (144R-63)u^3 + 158u^2 + 168u + 64, \\ b_2 &= 36u^3 + (216R-189)u^2 - 316u - 168, \\ b_3 &= 54u^2 + (108R-189)u - 158, \end{aligned}$$

and

$$\begin{aligned} c_1 &= 18u^5 + (54R+45)u^4 - (288R^2-378R+251)u^3 \\ &\quad + (948R-1086)u^2 + (1008R-1384)u + (384R-576), \\ c_2 &= (153R-72)u^4 - (657R-702)u^3 - (432R^2-1292R+1069)u^2 \\ &\quad + (2064R-2508)u + (1072R-1512), \\ c_3 &= (180R-108)u^3 - (891R-864)u^2 - (216R^2-1450R+1385)u \\ &\quad + (1116R-1422), \\ c_4 &= (6R-4)u^2 - (33R-32)u - (4R^2-58R-236337691420383). \end{aligned}$$

For $N \ge 0$, define

$$\mathcal{M}_N = M(0)M(1)\cdots M(N-1),$$

with the convention that $\mathcal{M}_0 = I$.

Choose initial conditions

$$A = \begin{pmatrix} A_1 & A_2 & A_3 & A_4 \\ B_1 & B_2 & B_3 & B_4 \end{pmatrix},$$

where

$$\begin{aligned} (A_1, A_2, A_3, A_4) = (&371693057604422527614 41,\ 111507917281327441564208, \\ &111507917281327599720129,\ 371693057604424109173 62), \end{aligned}$$

and

$$(B_1, B_2, B_3, B_4) = (1167416361542639692320,\ 3502249084627896132160,\\ 3502249084627879697280,\ 1167416361542622723840).$$

Write

$$A\mathcal{M}_N = \begin{pmatrix} P_{N,1} & P_{N,2} & P_{N,3} & P_{N,4} \\ Q_{N,1} & Q_{N,2} & Q_{N,3} & Q_{N,4} \end{pmatrix}.$$

Then, for each $j = 1, 2, 3, 4$*:*

$$\lim_{N\to\infty} \frac{P_{N,j}}{Q_{N,j}} = \frac{\sqrt{10005}}{\pi}.$$

B THE CHALLENGE CONJECTURES

This section contains open problems, presented here for the first time. In each case, the statement is written in the form of an explicit formula so that it can be checked numerically.

Conjecture 1 (An integral over knot polynomial roots expressing π^2)**.** *Let* $A_{7_2}(M, L)$ *denote the A-polynomial for the prime knot* 7_2*.*

$$\begin{aligned} A_{7_2}(M,L) = {} & L^5 + L^4\left(M^{14} - M^{12} + 3M^4 + 4M^2 - 2\right) \\ & + L^3\left(-2M^{18} + 5M^{16} + M^{14} - 4M^{12} + 6M^8 + 5M^6 + 2M^4 - 4M^2 + 1\right) \\ & + L^2\left(M^{22} - 4M^{20} + 2M^{18} + 5M^{16} + 6M^{14} - 4M^{10} + M^8 + 5M^6 - 2M^4\right) \\ & + L\left(-2M^{22} + 4M^{20} + 3M^{18} - M^{10} + M^8\right) + M^{22}. \end{aligned}$$

Let $\alpha \approx 0.349269\ldots$ *denote the real solution of* $A_{7_2}\left(\alpha, \alpha^{1/2}\right) = 0$ *closest to* 0.349269*. Let* $\beta \approx 0.406813\ldots$ *denote the real solution of* $A_{7_2}(\beta, \beta) = 0$ *closest to* 0.406813 *Let* $y = y(x)$ *denote the curve satisfying* $A(x, y(x)) = 0$ *for* $\alpha \le x \le \beta$ *and such that* $y(x)$ *is positive and decreasing with* $y'(x) \le -2$*. Prove the following formula, which was discovered empirically:*

$$\frac{4\pi^2}{85} = \int_\alpha^\beta \left(\log x \frac{dy}{y} - \log y \frac{dx}{x}\right).$$

This result is a surprisingly simple conjectural closed form of a Godbillon–Vey type Knot invariant (Khoi, 2008).

Conjecture 2 (Optimality of Apéry's irrationality-measure bound for $\zeta(3)$)**.** *The sequences* $(a_n)_{n\ge 0}$ *and* $(b_n)_{n\ge 0}$ *are solutions of Apéry's recurrence*

$$(n+1)^3 u_{n+1} - \big(34n^3 + 51n^2 + 27n + 5\big)u_n + n^3 u_{n-1} = 0 \qquad (n \ge 1),$$

with initial values

$$a_0 = 0, \quad a_1 = 6, \qquad b_0 = 1, \quad b_1 = 5.$$

Define $d_n \coloneqq \mathrm{lcm}(1, \ldots, n)^3$*. Note that* $d_n a_n, d_n b_n \in \mathbb{Z}$ *as shown in (Apéry, 1979).*

Prove:

$$\gcd(d_n a_n, d_n b_n) = e^{o(n)}.$$

The a_n, b_n sequences were generated by Apéry in his proof of the irrationality of $\zeta(3)$ (Apéry, 1979). Specifically, considering the linear forms $a_n - \zeta(3)b_n$ gives a bound on the irrationality measure of $\zeta(3)$, by demonstrating that $d_n a_n, d_n b_n \in \mathbb{Z}$.

The above conjecture shows that the irrationality-measure bound found by Apéry is tight for his particular formula.