AI in Mathematical Research

André Guerra

My 25-year-old self would have been crushed by what we are witnessing today. Nothing I write can reassure someone just starting a career.

Ken Ono, Guardian opinion piece

I am **in no way** an expert in AI.

Since May 2025 I have been using ChatGPT almost daily for research purposes, which is perhaps longer than the average, so I thought I would share my experiences.

My experience

In most of my PhD and early postdoc I studied something called **quasiconvexity**, but I partially abandoned this research direction because it was too difficult and I was stuck.

In July I was contacted by a friend at Google DeepMind, who knew almost nothing about quasiconvexity but had obtained several striking results on the topic.

We wrote a short paper together and I decided to use AI to go deeper in that direction, focusing on the **Iwaniec conjecture**, which is a central conjecture in the field with deep connections to Harmonic Analysis, Probability, Complex Analysis and PDE.

July-August 2026: a first partial result

AI disclosure

The author acknowledges the use of AI tools in preparing this paper, as the arguments here were developed in a dialogue between the author and ChatGPT-5.6 Pro.

The starting point for this paper originated in the author's earlier work [28] with R. Tione, and the intuition that, given the simple form of L, it should be possible to give a geometric interpretation of the quasiconvexity inequality for L through Morse Theory, as is done in [42]. A further intuition, which the author learned from many discussions with D. Faraco and L. Székelyhidi, is that L is closely related to separation properties of sets in $\mathbb{R}^{2\times 2}$, and one should see it as a distinguished member of the larger family described in Section 3 in order to find the correct proof. It is this viewpoint which clarifies the role of the matrices $\pm\mathrm{Id}$ in (1.2).

The author then fed [28, 42, 45] to ChatGPT and, through several prompts, it arrived at an outline of the proof described in the introduction. From the author's point of view, the main novel ideas brought forth by the AI were the simple but key identity (1.2)—which, however, was not expressed in terms of the $\det^{+-}, \det^{++}$ integrands, making it quite hard to parse—and in pointing the author to the existing literature on separating saddle points. The concise formulation of the required results from Morse Theory, described in Section 2, was obtained by the author essentially without AI assistance, and the details of the rest of the argument were filled by the author with some AI assistance. The author then generalized the proof described in the introduction to the whole family of separating integrands described in Theorem 3.3.

The text in this paper was written and checked entirely by the author, who takes full responsibility for its correctness.

September 2026

- 3 September: OpenAI releases publicly GPT-6 Astra.
- 5 September: G. prompts Astra:
  Check [G. 2026] but don't use the Internet otherwise. This is a challenge for you: can you prove quasiconvexity of L in $\mathbb{R}^{2\times 2}$?
- 7 September: G. sends the proof to a group of experts
- 8 September: The Navier–Stokes controversy begins
- 22 September: Hormozi, Kovac and Volberg get a proof of the Iwaniec conjecture
- 23 September: H. Yang (a 1st PhD student in Fudan) also obtains a proof
- 24 September: A first version of the proof appears on arXiv

It seems that all 3 proofs vary in detail and length but are almost identical in essence.

Some highlights from this summer

- 3-5 Aug: 2 counter-examples to Schiffer's conjecture (**97 y.o.**)
  [A third convex counter-example on 29 September.]
- 25-26 Aug: 2 counter-examples to Nevanlinna's half-plane problem (**101 y.o.**)
- 8-9 Sep: 3 proofs of Stein's discrete Euclidean-ball maximal problem (**30 y.o.**)
- Oct-Nov: 4 proofs of non-uniqueness for 2D Euler with vorticity in L^p (**63 y.o.**)
- 17-18 Nov: 2 proofs of the Pólya–Szegő polygonal torsion conjecture (**75 y.o.**)

Sharp Higher Integrability Theory, Part I: The L^p-norm of the Beurling–Ahlfors transform

Andrea Agazzi, Kari Astala, Rodrigo Bañuelos, Giuseppe Bruno, Gabriele Cassese, Daniel Faraco, André Guerra, Mahdi Hormozi, Bernd Kirchheim, Aleksis Koski, Vjekoslav Kovač, Jan Kristensen, Federico Pasqualotto, István Prause, Eero Saksman, Vladimír Šverák, László Székelyhidi, Riccardo Tione, Alexander Volberg, Hang Yang

Abstract

We confirm a conjecture posed by T. Iwaniec in 1982: the L^p norm of the Beurling–Ahlfors transform is $p-1$ if $p \geq 2$, and $(p-1)^{-1}$ if $1 < p \leq 2$.

AI disclosure

The key ideas of this paper were generated by ChatGPT 6.0 Astra, without substantial human assistance. The proof presented here does not belong to a single author, but rather to a community, imperfectly represented in the list of authors of this manuscript, whose efforts across the last four decades make the solution of this beautiful problem finally possible.

Mathematics as an experimental science

Experimental science

1. Formulate a research question.
2. Design an experiment.
3. Probe nature using tools.
4. Interpret the data inside a theory, and use it for theory-building.

AI-assisted mathematics

1. Formulate a research question.
2. Design a prompt.
3. Probe the problem using tools.
4. Interpret the output inside a theory, and use it for theory-building.

The human component is to **ask the right questions** and **interpret the answers**.

But mathematics differs from the experimental sciences:

- **Verifiability replaces reproducibility:** the output can be checked without access to tools or to details about how the output was obtained.
- **FAST replaces slow:** proofs can be generated in the matter of hours.

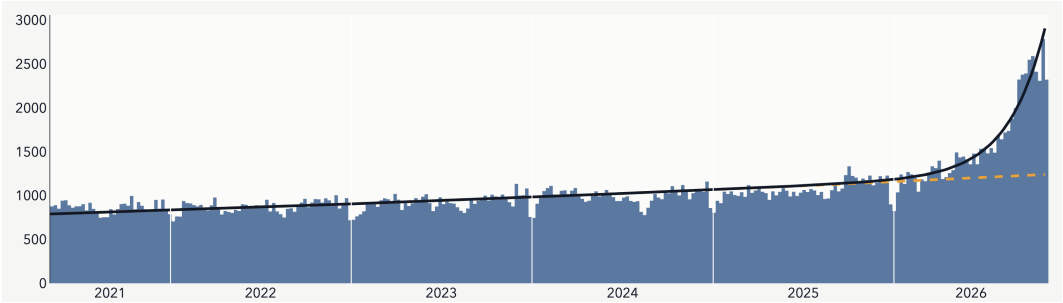


Figure: Number of daily arXiv papers in math, from `jaume.dedios.cat/exponentials`

Access to powerful models is necessary

AI is now far better than any human at coming up with proofs.

Thus, in the same way that it is impossible to do experiments without suitable equipment, it is now **impossible** to do cutting edge research without access to very powerful AI.

Disclosure: for this academic year I have free access to the best commercially available ChatGPT model as part of OpenAI's program for Academic Researchers.

Presently, we have:

- Top tier models: ChatGPT and Claude.
- Google models: somewhat behind, maybe 3 months.
- Open source models, such as SAIR (which DM provides): about 1 year behind.

However, the open source models will improve soon, and it seems clear that paying for very expensive models is not sustainable long-term.

Other institutions

In Cambridge, the department is reimbursing up 90 GBP per month to students and postdocs. Research grants can also be used to cover costs.

This information is not publicly available, and it's hard to find public information, but from what I could gather:

University	AI access
MIT	GPT Plus, GPT-6 Pro for 250 users
Duke	ChatGPT Edu (Plus-level); Claude
Princeton	GPT-6 Pro
Oxford	ChatGPT Edu (Plus-level)
UCLA	ChatGPT Edu (Plus-level)
Imperial	GPT-6 Pro; Claude Sonnet/Opus

Table: Free institutional access to AI models (November 2026).

Many institutions (incl. above) also offer **discounted** access to more powerful models.

Beyond proof generation

Besides generating proofs, AI can also speed up drastically most of the other aspects of scientific production, specially through **coding agents**, which are capable of:

- **Auto-formalization.** E.g. the human formalization of Fermat's Last Theorem was started in Nov. 2023, while Claude took 11 days in Aug. 2026 to do it.
- **Coding and numerics.** AI can be used for very fast implementation of algorithms and debugging, and allows the user to work with unfamiliar computational libraries.
- **Literature search.** AI is useful in discovering relevant literature and in doing quick literature comparisons, e.g. between results or approaches.
- **Writing documents.** The generated text is usually of poor quality (for now!!), but technical elements, such as proofs, can be outstanding.
- **Preparing slides.** Disclaimer: I typed every text line in these slides, but most latex code was written by AI.
- **Reviewing.** AI can help the reviewer in quickly finding flaws in proofs.

Tsimerman estimates AI yields a **10x productivity factor**.

An industrial setup for doing mathematics?



YouTube: Hyperoptimized Engineer 2026

The consequences of enhanced productivity

Using a powerful AI can feel **exhilarating** (I can now do everything I was doing before, except 10x as faster and much better!) but can also feel **stressful and exhausting**.

Diverging Limits Episode 2 podcast, with Scott Armstrong.

Interviewer: Do you find yourself going crazy?
SA: Yes. Yes, I can confirm that. So, I wear an apple watch and track my sleep because I'm that kind of person. And my sleep is like 1 hour and 7 minutes less in 2026 than in 2025, which is a lot. [...] I used to have no problems with sleep and now I just wake up every night at 3:30 a.m. like a crazy person. [...] It's driving me crazy.

Tentative advice: general advice

Caveat: given the pace of change in technology, this advice may be quickly obsolete.

- **Find a workflow that works for you.** Experiment with different levels and modes of using AI until settling on something sustainable.
- **Keep a research log.** Keep a log of failed routes, useful information (identities, lemmas, etc), and which of this information you have verified.
- **Experiment with coding agents.** More on that below.

Tentative advice: using LLMs in research

- **Build a prompt library.** Keep a list of reusable prompts for repeated tasks, e.g. checking the correctness of a proof, comparing papers, etc.
- **Give the right context**. Provide a concise document with the key information relevant for a given task.
- **Separate exploration from verification.** It's best to start by generating ideas freely and then, once you believe you have the sketch of a proof, give it to a new LLM session without the previous discussion and ask for an adversarial review.

Tentative advice: Coding agents

- **Create a rules file** for rules that apply to essentially all tasks.
- **Rule example: notation.** Coding agents have a strong tendency to "improve" notation locally, e.g. writing $s = x^2, r = |x|$, etc. Write a NOTATION.md file with guidelines, and then give instructions to avoid introducing existing notation.
- **Give concrete tasks.** For instance, "Improve Section 6" is a bad instruction. Better: "remove all unnecessary notation from section 6 and ensure every displayed equation needed later has a label".
- **Give examples.** Create a "style examples folder" in your project with polished passages from your papers that you'd like the agent to imitate linguistically and stylistically. Then add a rule for the agent to inspect those files and match their level of detail, sentence structure, formatting, etc.
- **Use version control.** Use Git and commit before any major changes.