LARGE LANGUAGE MODELS IN MATH

AARON PIXTON

1. Introduction

This document is a compilation of some of my current (August 2025) views on using LLMs (large language models, e.g. ChatGPT) in mathematics. I've been experimenting with them for the last three years and they keep getting better and better at solving certain types of problems. For example they now can solve a decent number of Olympiad problems and most algebraic geometry homework problems. I wanted a document that I could share containing my assessment of how this currently translates into actual utility (for me at least).

1.1. **Quick summary.**

I find LLMs helpful for:

- writing code to test or visualize examples;
- literature searches for possibly relevant papers to read;
- LaTeX assistance (making diagrams, transcribing notes).

I find LLMs unhelpful for:

- coming up with proofs;
- reading papers;
- directly teaching me math (as opposed to linking papers).

The main general principle at work here is that LLMs are a lot more useful in situations where you can easily verify the quality of their output. Compiling LaTeX or running code on test cases and seeing whether the results are what you wanted is easy, checking whether mathematical statements are true or false is hard.

2. Specific comments on different tasks

I'm just going to go through the bullet points from the above summary.

Date: August 2025.

2.1. **Writing code to test or visualize examples.** When I'm thinking about math I love working out concrete examples. I do this by hand when that's feasible, but I'll also often write simple one-off Python code to assist me for examples with more involved computations (e.g. to evaluate sums over combinatorial data or manipulate formal power series expansions). LLMs are very good at saving me time for this now - I just tell it I want Python code that accepts certain command-line arguments and does certain things with them, and it handles all the boring details (and I can check against smaller examples I've already done by hand). This saves a lot of time and also protects me from laziness when I'm not really in the mood to spend 30 minutes writing something myself that I'll never use again. I'm also a lot more likely to look at a geometric visualization of something if I can just ask the LLM to handle the technical details of graphing it.

2.2. **Literature searches for possibly relevant papers to read.** One of the main improvements with the most recent models (e.g. GPT-5-Thinking) for me is that they are pretty good at suggesting relevant papers on any given topic. I view this as some sort of semantic literature search, which is an improvement on keyword searches in many situations. For example, recently I was thinking about some algebraic geometry that suggested that given a finite connected graph, there should exist some sort of natural vector space (consisting of polynomials in variables corresponding to the vertices in the graph) of rank equal to the number of spanning trees in the graph. This would normally be hard to search for, but an LLM was able to give me a few potentially relevant papers to read (which turned out to include exactly what I wanted).

It's important here that you shouldn't trust LLM summaries of the contents of the papers it suggests - those will very often have inaccurate details. You still need to read the papers yourself.

2.3. **LaTeX assistance.** I've been using LLMs to help me make occasional TikZ diagrams in LaTeX for a couple years now - it is the perfect task for LLMs because you can immediately compile the LaTeX and check whether it looks like what you want. Usually there are some details that aren't great (e.g. labels on top of edges in graphs) and I have to follow up with requests to change things (or tweak parameters myself), but it still saves me a bunch of time compared with needing to remember or look up the appropriate TikZ syntax every time.

I've also done some experimentation with transcribing my handwritten notes into LaTeX and recent LLMs do a pretty good job of that now. They still get the wrong symbols sometimes or mangle commutative diagrams a bit, so you do want to check it against the source yourself. It used to be that LLMs would alter your sentences a lot (which made this unusable for me) but now they seem capable of preserving your precise language.

2.4. **Proofs.** Despite large advances, LLMs still don't seem that close to being useful to me for proving things. There are two real issues. The first is that verifying whether a natural-language proof is correct is very hard work under the best of circumstances, and LLM proofs are far from the best of circumstances. A human proof will usually have just a few points of likely error, where some key assertion is made and there isn't a clearly ironclad justification. These points can be difficult to check but at least it is clear where there might be an error. But with an LLM proof every single line is potentially load-bearing and suspect. A favorite LLM trick is to add an assumption out of nowhere that trivializes a problem. An example I saw recently involved the LLM breaking into cases based on whether an integer n was odd or even. Inside the even case, the LLM just casually included at some point "Since $n/2$ is odd," and never considered that n might be divisible by 4.

The second issue is that the LLMs only use very standard techniques in uncreative ways. They don't ever seem to come up with ideas tailored for a specific problem. This doesn't mean that they are incapable of proving useful results - many subresults in papers are proved in standard ways! But it does limit the utility of asking an LLM to prove something. At best you are getting a proof that isn't that interesting but that you still have to check extremely carefully. At worst you are wasting a lot of time reading incorrect proofs.

It might be that for some people working in some areas of math, asking an LLM to prove something that you suspect has a proof using standard methods is a reasonable thing to do. But it hasn't seemed very close to being very useful to me personally. I've also encountered several examples of mathematicians posting LLM proofs on social media as things that they were impressed by, then those proofs turning out to be bogus. I suspect most people are not as good at checking whether an LLM proof is correct as they think they are.

2.5. **Reading papers.** I haven't found LLMs particularly useful for summarizing math papers for me. They can provide a decent section-by-section outline of roughly how the paper is structured, but most papers will have such an outline in their introduction already. For anything more detailed, they seem to make too many mistakes - for instance you shouldn't trust that they are correctly restating any results from the paper. If it is a long paper and it isn't clear from the introduction where some issue is addressed, treating an LLM as semantic search inside the paper can be okay, but generally I'd rather just skim the paper myself.

Also, LLMs aren't currently useful for finding errors (beyond typographical/grammar ones) in math papers. It would be very surprising if they were given that they make a lot of errors themselves! I've heard that they can be useful for providing feedback on the clarity of your writing, but I haven't tried this myself.

2.6. **Learning mathematics.** My experiments with talking with LLMs directly about mathematics (as opposed to just looking at papers they reference) have been unsatisfactory enough that I'd advise extreme caution with reading what they write about any area that you don't have significant knowledge about already. When asking LLMs about topics that I do know a lot about, my typical experience is that they do well (but not perfectly by any means) with initial questions on a subject, reciting relevant standard definitions and stating mostly true facts about them. But then if I ask a follow-up question or two, the responses quickly become totally confused, conflating different definitions and making nontrivially false claims very confidently.

The way I view talking with current LLMs is that they are extremely well-read but lack deep understanding and are incapable of carrying on a real back-and-forth conversation about anything technical. This unfortunately makes them ill-suited as teachers.