LARGE LANGUAGE MODELS IN MATH

AARON PIXTON

1. Introduction

This document is a compilation of some of my **current** (August 2026) assessments of how useful LLMs (large language models, e.g. ChatGPT) are for my research math activities. I am not making any predictions about the future - this is just a snapshot of some of the ways I currently use LLMs for math, and how useful I find them. Also, I expect that this sort of thing is highly idiosyncratic - different people work in different ways on different types of math problems, so will find LLMs to be of greatly varying utility.

If you are interested in how my views have changed over time, a similar document that I wrote in August 2025 is available here.

1.1. **Quick summary.**

I now use LLMs fairly frequently for a variety of things, but I don't actually think they are accelerating my research by more than 15% or so. The main bottleneck is still my personal understanding of the mathematics involved, which is largely just a function of how much time I have spent directly thinking about it on my own or in conversation with other mathematicians.

Much of the utility to me comes from what I'll call *technical assistance*: things like speeding up writing code (to explore examples) and exploring the literature (locating potentially relevant papers to read). But those activities have never made up a particularly large proportion of the time I spend on mathematics, so the impact of speeding those up is relatively minor. What would be more impactful is if LLMs were good at directly helping me understand math better (e.g. by helping me (dis)prove things, making useful observations, or explaining perspectives with which I am less familiar). I think 2026 is the year in which some interaction with LLMs crossed the line to become net positive for my mathematical understanding, but their impact in this respect is still relatively small.

Date: August 2026.

2. Technical assistance from LLMs

The most straightforward way in which LLMs are useful to me is coding. Instead of doing extremely standard coding in Python/Sage myself to compute some example that is too messy to do by hand, I can just open up an LLM coding agent and explain in natural language what I want it to compute, and it will handle it (and follow-up requests) just fine in a fraction of the time. I still need to sanity check the results, but no more than I would need to for my own code. This really feels like nothing much has changed except that I've saved a bit of time that I can spend thinking about the math instead of about uninteresting implementation details.

There is similarly little to say about using LLMs to speed up LaTeX formatting and diagram creation. Instead of needing to struggle through TikZ syntax myself, I can just ask an LLM to make a diagram (with TikZ or Python). A workflow that I've used when making slides for a talk that has worked pretty well is to write a paragraph of text for each slide and include formatting instructions in natural language in brackets (including descriptions of diagrams to create in TikZ or Python), then run the whole thing through a coding agent to handle all the formatting at once.

I view literature search as another relatively mechanical task where LLMs can save a decent amount of time. If I'm not familiar with the literature in an area, then semantic search is often a better starting point than keyword search. For example, in August 2025 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 semantic search was very effective at immediately getting me very close to what I wanted. In my experience LLM literature search is far from infallible, but it is a nice quick way to find papers that are at least potentially relevant to any given topic, rather than having to stumble around the literature a bit first. It is just a starting point though - I still need to follow citations and such myself from there.

3. Improving mathematical understanding with LLMs

I'll discuss three ways to try to use an LLM to understand math better. First, there is LLM as explainer - you can give it someone else's mathematical writing (e.g. a paper) and ask it to explain it to you, maybe from a specific perspective. Second, there is LLM as sanity

checker - you can give it your own mathematical writing (e.g. a couple paragraphs describing a statement that you don't fully understand) and see if it notices something wrong with your setup or can easily find a counterexample or a simple/standard proof. Finally, there is LLM as problem solver - you can give it a problem that you expect is hard, and hope that its response will help you solve the problem in some way (or even that it will fully solve the problem itself).

In general I find LLMs to be poor explainers and good sanity checkers. They haven't been particularly useful as problem solvers for my research so far.

3.1. **LLMs as explainers.** I have not had much success with the first of these options. LLM mathematical writing tends to oscillate between two poles: either it is bloated with unimportant details (with no indication of which details are the important ones), or it is summarized down to the point of being incorrect or meaningless. (I actually think reading LLM summaries of papers is one of the most dangerous things you can do - you'll end up missing critical hypotheses in the main results or totally misunderstanding the general setup. I'd rather just skim the introduction.) The only real way in which I've found LLMs helpful for reading a paper is by helping me resolve my confusion about something after reading the paper, rather than by directly explaining the paper itself. I think this is more properly interpreted as a form of sanity checking though.

3.2. **LLMs as sanity checkers.** This option has worked better for me. The basic setup is that I am thinking about something on my own and reach some mathematical statement that I don't fully understand. Maybe it's a statement where I'm really not sure whether or not it is true, or maybe it is a statement that I strongly suspect is true by some standard argument but I've forgotten the argument (an example from earlier this year - any convex polytope admits a triangulation that doesn't create any new vertices). Maybe there are two statements that I thought were both supposed to be true, but they seem to contradict each other and I haven't figured out how to resolve the contradiction yet. There are a lot of possible scenarios, but the common feature is that I've been thinking about something on my own and I want to know if I'm missing something simple before continuing. Sometimes it is a statement that I've been poking at for hours, or sometimes it is a statement that I've thought about for only a few minutes.

There is a nontrivial time cost to ask an LLM such a question. I need to carefully write down a self-contained description of the statement/problem, which can be quite involved - frequently I'm fairly far

down some rabbit hole and have all sorts of specialized setup and conventions that need to be specified for the statement to make sense. And then I need to wait for an indeterminate time (up to 3 hours sometimes) for a serious reply from a frontier LLM. It doesn't generally make much sense to do this if I'm just going to spend a bunch of that time actively thinking about the problem myself, so I'm more likely to ask an LLM about something if I'm about to get up and take a break anyway (or at least am ready to switch to thinking about some other aspect of my research instead).

The good part of all this is that carefully writing up a subproblem description is a worthwhile thing to do anyway to make things clearer in my own mind (often just doing the writing makes things clearer to me and the LLM is unneeded) and taking breaks is also a good thing to do. So the cost of making this sort of LLM query is still fairly low. This is good because the reward is also relatively low. Checking the well-posedness/plausibility/obviosity of a statement is nice, but of course I'm already trying to check those things myself as I go (and still will be, independently of the LLM).

This type of sanity checking use has the virtue that it doesn't take long to glance at the LLM response and see whether it noticed something. If the LLM instead claims that something is true but needs a more involved argument, maybe involving deep results from the literature, then I'm in the position of needing to check the LLM instead of the other way around, and that is not necessarily worthwhile.

3.3. **LLMs as problem solvers.** So far LLMs have not been particularly useful to me for directly solving the problems that I care about. There are many widely publicized examples at this point of LLMs solving important problems with seemingly minimal human involvement - two of my personal favorites are the disproof of the Erdős unit distance conjecture and the construction of a complex structure on S^6. But these impressive examples obscure that the base rate of success is quite low on problems that one has reason to believe are hard. And certain types of problems (and areas of math) seem to be much harder for LLMs than others - for instance many of the problems I work on are of a more open-ended form like "come up with a good formula for X" or "find a nice description of the cohomology of Y" rather than simply "prove or disprove P", and LLMs seem to have some trouble with this.

This is not to say that I think it is useless for me to try asking LLMs to solve problems that interest me. But for now, if I expect a problem is hard and am asking an LLM about it, it is usually not because I hope to receive a full solution. Indeed, if an LLM claims to have a

lengthy proof, I might not even try very hard to understand or verify the argument! Verifying whether a natural-language proof is correct can be very hard work under the best of circumstances, and LLM proofs are far from the best of circumstances. And often the things I am asking an LLM about are not of very much intrinsic interest - they are narrow side questions that I arrived at while musing about a different, more important problem. They are also probably the "wrong" questions to ask. I'd like to understand them better, but it isn't necessarily worth my time to wade through too much LLM writing, especially when there is a high chance that the claimed proof has gaps in it. So if an LLM produces many pages of poorly written mathematics and it isn't immediately clear to me whether there are any interesting ideas, I might leave it at that.

Instead, I am asking the LLM partly for sanity checking (as in the previous section) and partly in the hope that I'll notice something interesting in the LLM's writing about the problem - maybe it is looking at the problem from a different perspective, or maybe it makes one small observation that I hadn't noticed. In a recent instance of this, I asked an LLM to try to prove a statement using a specific approach that I suggested to it. It produced a claimed proof with four nontrivial steps. The first step had a small observation about the problem that I had not yet noticed. The remaining three steps each involved claiming very believable lemmas with extremely sketchy or incorrect justifications. Because I actually cared a fair amount about proving the statement in this case, I spent a few days trying to understand these lemmas (and trying to get the LLM to fix its arguments for them) before giving up on the LLM's approach and just coming up with a better argument myself to replace those three lemmas. In this case, I would have been better off just noting the useful observation in the first step and not trying to engage with the remainder of the claimed proof.

In my experience it has been rare to get something as concrete as a useful observation like that from LLM proof attempts. I think it is worth my time to make queries like this, but primarily just because seeing the first paragraph or so of an attempt on a problem might show me a different perspective on what is going on mathematically, not because the LLM is directly solving the problem for me.