Evolution of ChatGPT in Mathematica: a user experience

Galina Filipuk and Andrzej Kozłowski

This paper explores the impact of artificial intelligence on research and education, using ChatGPT and Mathematica as case studies to illustrate AI's influence across both fields.

When ChatGPT by OpenAI[1] [2] debuted in 2023, it almost overnight became a worldwide sensation. It seemed to be able to engage in remarkably human-like conversations on a wide range of topics, about which it seemed to have extensive knowledge. However, concerns soon emerged about disconcerting reports of it making up facts, false quotations and inaccurate bibliographic references as well as "lying." This unexpected phenomenon was given the name "hallucinations." The idea of testing ChatGPT's skills in mathematics occurred to many, but it became clear that the results would be underwhelming, since chatbots are not designed to perform any computations, even basic arithmetic ones.

Given how Large Language Models (LLMs) operate, this outcome is unsurprising. LLMs are trained on huge collections of examples and generate text by predicting the "most likely word" that continues a given phrase. They cannot execute loops or branching, which are essential in most computational algorithms. Nevertheless, there have been reports of ChatGPT-4 writing adequate programs in certain programming languages. So it seemed interesting to determine whether it could be useful to mathematics students who wish to use Mathematica,[2] but are not skilled programmers. It turned out that a similar idea was explored earlier to Stephen Wolfram. First, ChatGPT gained the capability to access Wolfram|Alpha[3] [5], enabling it to return reliable answers to purely computational questions. More impressively, from version 13 ChatGPT became integrated directly within Mathematica, serving two roles: as "Wolfram Language Assistant" and as "Code Writer." It now seems possible that ChatGPT with this new ability could help lower Mathematica's learning curve, and perhaps even to solve mathematical problems, at least those that can be solved with the help of Mathematica.

This note aims to describe the findings of our investigation into the potential of combining ChatGPT with Mathematica for teaching analysis to students with little or no prior knowledge of Mathematica. We believe that some of our conclusions may be both surprising and intriguing. However, a degree of caution appears advisable. GPT and other AI chatbots are evolving extremely fast. Even during our early experiments with ChatGPT, we noticed changes in ChatGPT's behavior, often improvements, but occasionally regressions. With the proliferation of GPT models and versions, keeping track of their features and quirks has become increasingly difficult. In addition, many chatbots beyond GPT have appeared, which were added to later versions of Mathematica. More recently, Wolfram developed its own model, which is now the default in any "chat-enabled" notebook. However, using these models from within Mathematica is not free. When ChatGPT was the sole option available, we obtained a grant to carry out this investigation. For this reason we have not experimented other models for this investigation, although we believe the conclusions would largely hold [1, 3, 4, 6]. Wolfram's own model looks particularly promising, as it was designed specifically for Mathematica.

Another natural question is what would happen if we replace Mathematica with other similar programs. We have not investigated this, because we specifically focused on using Mathematica for teaching analysis at the University of Warsaw. Also, one of our key conclusions is the importance of chatbots being able to perform their own evaluations, which allows them to identify and attempt to correct their own errors. When we asked GPT about this, it acknowledged having some familiarity with other programming languages, but explained that it cannot carry out its own evaluations in them. However, for those interested in using AI with other systems there are various resources available online.[4]

Our approach involved asking ChatGPT to solve, with the help of Mathematica, undergraduate analysis problems, mostly taken from classes we have taught. In our early experiments, we were often confused by the fact that ChatGPT's behavior varied over

[1] https://openai.com
[2] https://www.wolfram.com
[3] https://www.wolframalpha.com

[4] For instance, https://www.maplesoft.com/products/maple/new_features/ or https://www.mathworks.com/discovery/chatgpt.

time, as well as between different models and "personas" that one could choose in a Mathematica notebook. One problem was that sometimes ChatGPT was able to evaluate the Mathematica code it wrote, while other times it could not. Additionally, it occasionally provided incorrect answers to a computational problem, claiming they were produced using Mathematica. When we pointed out discrepancies between ChatGPT's responses and Mathematica's actual output, it would apologize but fail to explain why it had not properly used Mathematica. When ChatGPT was unable to evaluate Mathematica code, its abilities were seriously reduced. We eventually discovered that ChatGPT's ability to perform its own evaluations could be enabled or disabled in the "Add and Manage LLM Tools" menu that appears in the top right corner of a chat-enabled Mathematica notebook. After disabling this feature, specifically the WolframLanguageEvaluator, we repeated some of our earlier queries and observed notable differences in the responses. Wolfram Language, the programming language of Mathematica, is built for computation. Like other languages it is made up of "words," but each word corresponds to a specific computational task (detailed in the documentation). By combining these words, one can, in principle, carry out every computation, subject to the limits imposed by computational complexity. Programs written in ordinary programming languages have to follow strict syntax. Mathematica supports "free-form" input. This allows users to express queries in natural language, which Mathematica then attempts to interpret and convert into a computationally meaningful form. This means that in principle one can simply ask Mathematica to perform a computation without needing to learn the intricacies of Wolfram Language.

1 Curvature of a surface

When one asks Mathematica, via free-form input, to compute the curvature of a surface such as a sphere, it fails to return an answer. This is because Mathematica does not include a built-in function for this specific task, and free-form input is limited to referencing existing built-in functions. While there are several ways to compute surface curvature in Mathematica without additional programming, these methods are often difficult to discover for inexperienced users. This is precisely the kind of situation where one might hope ChatGPT could offer meaningful assistance. Asking ChatGPT to compute the curvature of a sphere using Mathematica leads to a surprising outcome. Although this is a relatively simple problem that ChatGPT can solve on its own, ChatGPT "invents" a function, sometimes called "Curvature," other times "GaussianCurvature," which it incorrectly claims is built in Mathematica. It then writes code that could indeed accomplish the task, if such a function actually existed. Needless to say, its code does not work.

```
Is there a way to ask Mathematica to compute the
Gaussian curvature of a surface? For example,
a sphere of radius r?
```

Yes, you can compute the Gaussian curvature of a surface using the Wolfram Language. For a parametric surface, you can use the GaussianCurvature function.

Here is how you can compute the Gaussian curvature of a sphere of radius r:

1. Define the parametric equations of the sphere.
2. Use the `GaussianCurvature` function to compute the curvature.

For a sphere of radius r, the parametric equations are:
$r(\theta, \phi) = (r\sin(\theta)\cos(\phi), r\sin(\theta)\sin(\phi), r\cos(\theta))$,
where θ ranges from 0 to π and ϕ ranges from 0 to 2π.

Here's how you can do it in the Wolfram Language:

```
r = r; (* Radius of the sphere *)
parametricSphere = (r Sin[θ] Cos[φ],
                    r Sin[θ] Cos[φ], r Cos[φ]);

GaussianCurvature[parametricSphere, (θ,φ)]
```

This behavior typically occurred when ChatGPT lacked the ability to evaluate its own code. In such cases, it was necessary to inform ChatGPT that its code did not work. With some guidance or carefully phrased hints, it could sometimes be steered toward a valid method for performing the computation. This approach required the users to already know the solution themselves in order to prompt ChatGPT correctly. In contrast, when ChatGPT was able to evaluate its code, it would initially follow the same path by generating a non-functional solution. But then, upon attempting to execute the code, it would recognize that the result was invalid. Remarkably, it could then consult the documentation and revise its approach, often coming up to a correct and working solution on its own.

2 Painlevé transcendents

A similar issue happened when we asked ChatGPT to compute Painlevé transcendents, which are not implemented in the current version of Mathematica. Each time, ChatGPT confidently asserted that these functions were available and even generated plausible sounding names for them, often different with each attempt.

```
Can one compute Painlevé transcendents with Mathematica?
```

Yes, you can compute Painlevé transcendents using Mathematica (Wolfram Language). The Painlevé transcendents are solutions to the Painlevé equations, which are certain nonlinear ordinary differential equations.

In Wolfram Language, the function PainleveP can be used to compute these transcendents. Here is an example how to use it:

```
PainLeveP[1,x]
```

This function computes the first Painlevé transcendent P_1.

To see the full details and options for this function, let me get the documentation for you.
It seems there was an issue with the lookup. Let me try again.

▶ Used **Documentation Searcher**

It seems that the search returned some general information about the Painlevé transcendents rather than specific functions in Wolfram Language. However, I can confirm that the function to compute Painlevé transcendents in Wolfram Language is indeed PainleveP.

Here's an example of how to use it:

```
(* Compute the first Painlevé transcendent P1 *)
PainLeveP[1,x]
```

Would you like to see computations or plots of specific Painlevé transcendents? Let me know how I can assist further!

Once again, when it was able to evaluate its code, ChatGPT recognized the error and identified an alternative way to compute the values.

```
So does Mathematica have a built-in Painlevé function or not?
```

The Wolfram Language does not have a built-in function that directly computes Painlevé transcendents in a closed form. However, you can solve the differential equations numerically using NDSolve as demonstrated.

It seems that there was an error in the replacement part of the code. Let me correct it and provide the accurate result for the value at $x = 1$:

Correcting the computation of the value at $x = 1$

```
(* Define the first Painlevé equation
p1Equation = y''[x] == 6y[x]^2+x;

(* Initial conditions *)
initialConditions = {y[0] == 0, y'[0] == 1};

(* Solve the differential equation numerically *)
p1Solution = NDSolve[({p1Equation,initialConditions},
                                                    y, {x, 0, 1}];

(* Compute the value of y at x = 1 *)
yValueAt1 = y[1] /. p1Solution[[1]]
```

Insert and evaluate   Insert   Copy

Let me evaluate this for you.

3 Computation of a limit

There were several other instances where ChatGPT's performance noticeably improved once it was granted the ability to evaluate its own code. We asked it to compute the limit of a sequence of real numbers defined by a simple nonlinear recurrence relation. Although ChatGPT was aware that Mathematica includes a function (RSolve) for solving recurrence equations, it noticed that the equation was nonlinear and assumed RSolve would not be applicable. As a result, it recommended a numerical approach instead. However, while RSolve cannot generally solve nonlinear recurrence equations, it is capable of handling this particular case. When ChatGPT was able to perform evaluations, it correctly used the built-in functions RSolve and Limit to get the correct answer. In fact, for the example we provided, the limit of the sequence can be determined without solving the recurrence relation, through a simple mathematical argument. ChatGPT did not initially recognize this. However, after computing the limit and observing that it was 0, it was then able to formulate a valid mathematical proof. This behavior is fairly typical for ChatGPT, and as human reasoning: once it knows the correct answer, it can often construct a plausible, and sometimes even correct, justification.

```
Consider the system of recurrence equations: a[1] == x, a[n+1] = a[n](99/100 + 10/n), where x is some real number. Can you use Wolfram Language to find the limit of the sequence a[n]?
```

To define the recursive equation, we can use the RSolve function. However, since RSolve cannot handle recursive equations with a variable coefficient, we'll need to use an alternative approach.

We conclude that ChatGPT, when used within Mathematica, can serve as a valuable tool for users who are not well acquainted with the Wolfram Language, helping them navigate and use it more effectively, provided, of course, that its outputs are approached with a critical mindset. For experienced mathematicians, this should not be a problem. However, the situation is quite different for students whose primary aim is to learn mathematics. In such cases, reliance on ChatGPT can be problematic, as it is not always a reliable mathematician. ChatGPT is quite good at stating general theorems and even outlining entire theories, but it is prone to errors in reasoning, often the kind that may appear convincing to a less careful student. For instance, we were able to persuade ChatGPT that "all positive integers are equal" using a well-known fallacious inductive proof. Although it initially accepted the argument, once we pointed out the obvious contradiction, ChatGPT acknowledged the mistake and was able to identify and explain the flaw in the reasoning. That said, ChatGPT's ability to use Mathematica for performing computations, writing simple code, and even constructing illustrative examples for mathematical concepts and theorems can still be of significant value to students.

Much like a diligent student, ChatGPT can often find a solution after receiving a hint and can then apply the same method to solve similar problems. Unfortunately, all of this progress is lost once the chat session ends or the allocated memory is exhausted. If ChatGPT could retain knowledge from these conversations, it could be trained on a wide range of mathematical problems. Combined with Mathematica's computational power, it might evolve into a powerful tool for mathematical education. We should note that ChatGPT has recently introduced a memory feature, allowing it to retain information from user interactions and apply it in future chats. Unfortunately, this functionality is currently limited to paying subscribers of ChatGPT Plus and does not support sharing learned information between users.

Generative AI models like GPT are evolving rapidly, and their performance is improving. Our recent experiments on GPT-4 Omni show that it can now produce correct, rigorous solutions to certain first-year undergraduate problems while generating and explaining appropriate Mathematica code. However, several persistent limitations remain: inventing plausible but non-existent Mathematica functions (such as a fictitious RadiusOfConvergence function for power series); sometimes showing an inexplicable reluctance to use Mathematica, resulting in incorrect answers to computations that Mathematica can solve accurately; and, on the mathematical side, favoring vague generalities over precise, detailed explanations and engaging in "information dumping," i.e., providing unnecessary and unsolicited information.

Despite these limitations, we believe ChatGPT, and similar AI models, can be genuinely useful in mathematics education if integrated with a reliable computational platform. Notably, its ability to generate code, and more importantly, to clearly explain complex code, combined with its ability to rapidly produce computational examples that illustrate mathematical concepts, could substantially enhance both teaching and learning experiences. Since our initial study, ChatGPT has shown marked improvement. For instance, ChatGPT now correctly solves recurrence relation problems that previously stumped it.

```
Consider the system of recurrence equations: a[1]
== x, a[n+1] = a[n](99/100 + 10/n), where x is some
real number. Can you use Wolfram Language to find
the limit of the sequence a[n]?
```

To find the limit of the sequence defined by the recurrence $a[1] = x$ and $a[n+1] = a[n](\frac{99}{100} + \frac{10}{n})$, we can analyze the behavior of the sequence as n approaches infinity.

We can use DiscreteLimit function to compute the limit of the sequence. Here is how you can set it up in Wolfram Language:

```
DiscreteLimit[
  RSolveValue[{a[1] == x,
        a[n+1] == a[n] * (99/100+10/n)}, a[n], n],
  n→∞
]
```

Insert and evaluate   Insert   Copy

This will give us the limit of the sequence $a[n]$ as n approaches infinity. Let's evaluate this expression.

Used Wolfram Language Evaluator +

The limit of the sequence $a[n]$ as n approaches infinity is 0. This means that regardless of the initial value x, the sequence converges to 0.

The answer to the same question given by Wolfram's new LLM kit is almost identical.

4 Computing with OpenAI O1 model

The evolution of ChatGPT (and other LLM chatbots) is proceeding at a great pace. Only after completing the first version of this article we discovered that a new version of GPT, named OpenAI O1 model, appears to represent a quantum leap in LLM model's mathematical capabilities. The new version not only was able to almost flawlessly deal with all the computational problems which proved too demanding for previous versions but is often able to give rigorous mathematical proofs, which may be challenging even for advanced undergraduate students. Here is an example from complex analysis, which illustrates the computational and theoretical capabilities of this version of GPT.

```
We want to find a Möbius transformation f of the
complex plane, which is not the identity itself and
which preserves the real axis (takes real numbers
to real numbers) and is such that f(f(f(f(z))))=z,
that is, if we iterate f five times we get the
identity transformation. How can we use Wolfram
Language to do this?
```

Below is one illustrative way to have the Wolfram Language search systematically for real-coefficient Möbius transformations of exact order 5. The idea is:

1. Represent the Möbius transformation by a 2×2 real matrix M with $\det(M) = 1$.
2. Having $f^5 =$ Identity as a projective transformation means M^5 is (up to a nonzero scalar) the 2×2 identity matrix. In $SL(2, \mathbb{R})$, $M^5 = I$ exactly.
3. Solve for real (a, b, c, d) subject to $M^5 = I$ and $ad - bc = 1$.
4. Exclude the trivial identity solution $M = I$ (which would give $f =$ identity).

A suitable code outline is:

This is followed by Wolfram language code, which indeed finds all the solutions asked for. Still, even this version is not ready to replace human mathematicians. For some reason it seems unable to examine the result of evaluating its own code and so cannot see that it returns a number of non-trivial solutions. In fact, the existence of such solutions is obvious when one views Möbius transformations as projective transformations of the Riemann sphere: they are given by rotations by $2\pi/5$ along the great circle corresponding to the real line. But even so, its performance already far exceeds the expectations we had when we embarked on this project.

Because real coefficients mean that the real axis is mapped to itself, any nontrivial real solution here will give you a Möbius transformation preserving the real axis and satisfying $f^5 =$ identity. In many cases, you will find that the only real solutions are the $\pm$identity matrices (which correspond to the identity projective transformation), demonstrating that there is no nontrivial real-axis-preserving Möbius map of exact order 5.

Why might there be no nontrivial solution? In the group $PSL(2, \mathbb{R})$, elements of finite order >2 are "elliptic" transformations that typically do not leave the entire real axis invariant as a set. (They have a single fixed point in the upper or lower half-plane instead, and so their action on the real boundary is not an order-5 cycle.) Hence the system of equations can force the matrix to collapse to ± 1.

If you run the code above and do indeed get only the identity/negative identity solutions, that confirms there is no Möbius transformation (with real coefficients) of order 5 other than the identity itself.

▶ Key Points

- A Möbius transformation $f(z) = (az + b)/(cz + d)$ with real a, b, c, d preserves the real axis.
- Requiring $f^5 =$ identity means the associated 2×2 matrix M must satisfy $M^5 =$ scalar $\times$ I. In $SL(2, \mathbb{R})$, that means $M^5 = I$.
- For finite orders >2, real-coefficient 2×2 matrices in $SL(2, \mathbb{R})$ typically fail to fix the whole real axis, so the only solutions end up being ± 1.

You can experiment with similar code (perhaps relaxing the real-coefficient assumption) to discover Möbius transformations of higher finite order that do preserve some other curve or circle in the complex plane, but necessarily the real axis.

Acknowledgements

The authors are grateful to Professor Youcef Mammeri for help with preparing this article.

References

[1] N. Korkmaz Güler, Z. G. Dertli, E. Boran and B. Yildiz, An artificial intelligence application in mathematics education: Evaluating ChatGPT's academic achievement in a mathematics exam. *Pedagogical Res.* **9**, article no. em0188 (2024)

[2] OpenAI, ChatGPT (Large language model). (2025) https://chatgpt.com

[3] B. Pepin, N. Buchholtz and U. Salinas-Hernández, A scoping survey of ChatGPT in mathematics education. *Digit. Exp. Math. Educ.* **11**, 9–41 (2025)

[4] Y. Wardat, M. A. Tashtoush, R. AlAli and A. M. Jarrah, ChatGPT: A revolutionary tool for teaching and learning mathematics. *Eurasia J. Math. Sci. Technol. Educ.* **19**, article no. em2286 (2023)

[5] Wolfram Research, Inc., Mathematica (Software, version 14.2) (2025) https://www.wolfram.com/mathematica

[6] Y. Zhuang, Lessons from using ChatGPT in calculus: Insights from two contrasting cases. *J. Form. Des. Learn.* **9**, 25–35 (2025)

Galina Filipuk and Andrzej Kozłowski are the authors of a three-volume book on mathematical analysis with Mathematica published by de Gruyter. The second edition now in print contains a chapter on using ChatGPT with Mathematica.

Galina Filipuk is an associate professor at the Institute of Mathematics, University of Warsaw. She received her PhD in mathematics from the Belarussian State University in 2002. She worked as a research associate at Loughborough University (UK) and as a research assistant professor at the Polish Academy of Sciences. She hold Humboldt Research Fellowship for Experienced Researchers at TU Dresden and KU Eichstätt-Ingolstadt (Germany). She published more than 100 papers in special functions.

filipuk@mimuw.edu.pl

Andrzej Kozłowski is an adjunct professor at the Institute of Applied Mathematics and Mechanics, University of Warsaw. In 1968 he emigrated at the age of fifteen from Poland to the UK, where he completed his high school education and received his BA in mathematics from Warwick University and DPhil in algebraic topology from the University of Oxford. In 1980 he became a Japan Society for the Promotion of Science Postdoctoral Fellow at Tsukuba University (Japan). He worked as an assistant professor of mathematics at Wayne State University (USA) and as a professor at Toyama University of International Studies (Japan), and also taught at Tokyo Denki University (Japan). In 1993, he was a visiting professor at the University of Warsaw. He lived and worked in Japan for 25 years before returning to Poland in 2010. He has published over 30 research articles in algebraic topology and several in computer algebra and mathematics education. For several years he was a consultant for Wolfram Research, for whom he produced a number of Mathematica demonstrations.

akoz@mimuw.edu.pl