The AI Research Assistant: Promise, Peril, and a Proof of Concept

Tan Bui-Thanh, *Department of Aerospace Engineering & Engineering Mechanics, and Oden Institute for Computational Engineering & Sciences, University of Texas at Austin, Austin, TX, 78712, USA*

Abstract—Can artificial intelligence truly contribute to creative mathematical research, or does it merely automate routine calculations while introducing risks of error? We provide empirical evidence through a detailed case study: the discovery of novel error representations and bounds for Hermite quadrature rules via systematic human-AI collaboration.

Working with multiple AI assistants, we extended results beyond what manual work achieved, formulating and proving several theorems with AI assistance. The collaboration revealed both remarkable capabilities and critical limitations. AI excelled at algebraic manipulation, systematic proof exploration, literature synthesis, and LaTeX preparation. However, every step required rigorous human verification, mathematical intuition for problem formulation, and strategic direction. We document the complete research workflow with unusual transparency, revealing patterns in successful human-AI mathematical collaboration and identifying failure modes researchers must anticipate. Our experience suggests that, when used with appropriate skepticism and verification protocols, AI tools can meaningfully accelerate mathematical discovery while demanding careful human oversight and deep domain expertise.

The integration of artificial intelligence into scientific research has triggered a fundamental debate about the nature of scholarly work. Nowhere is this controversy more acute than in mathematics, a field that prizes rigor, creativity, and human insight. Can large language models (LLMs) like `GPT-4` and `Claude` meaningfully contribute to mathematical discovery, or do they merely produce plausible-sounding but potentially flawed outputs that mislead researchers?

This paper addresses these questions not through philosophical argument but through documented experience. The closest pure-math counterpart to our work is [1], though the latter does not provide the same level of detail, the patterns of success and failure, etc. Over the past two months, we conducted a systematic exploration of human-AI collaboration in mathematical research that produced novel theoretical results for Hermite quadrature error estimation. Rather than hide the AI's role, as might be tempting given current academic norms, we provide unusual transparency about the entire research process, including both successes and failures.

While systematic evaluations on benchmarks [1], [2], [3], [4] assess AI capabilities in controlled settings, our detailed case study documents the messy reality of using AI for actual research with all its iterative refinements, dead ends, and verification challenges. The broader AI math reasoning literature provides the context, while our case study provides the lived experience. Both are necessary for understanding the full picture.

THE CONTROVERSY: PROMISE AND PERIL

The use of AI in mathematics research has generated polarized responses. Optimists point to AI's potential to accelerate discovery. Recent work demonstrates diverse AI assistance approaches: machine learning systems identified patterns in knot theory that guided mathematicians to prove new theorems connecting

XXXX-XXX © 2026 IEEE
Digital Object Identifier 10.1109/XXX.0000.0000000

algebraic and geometric invariants [1], and formal theorem proving systems using reinforcement learning achieved silver-medal performance at the 2024 International Mathematical Olympiad [2].

Skeptics raise serious concerns. LLMs are known to "hallucinate" plausible but incorrect mathematical statements. They lack genuine understanding and cannot reliably verify their own outputs. Critics worry that over-reliance on AI may atrophy human mathematical intuition, introduce subtle errors that escape detection, or simply replace deep thinking with superficial pattern matching. For example, the authors of MATH benchmark [3], consisting of 12,500 competition mathematics problems, found that even the largest GPT-3 models achieved only 3–7% accuracy, and they concluded that "accuracy remains relatively low, even with enormous Transformer models." Critically, models exhibited severe overconfidence: GPT-2 assigned near-100% confidence to both correct and incorrect answers, achieving only 68.8% AUROC (Area Under the Receiver Operating Characteristic curve) in distinguishing them, which is barely above random chance at 50%.

A more relevant example for this paper is the comprehensive evaluation [4] of ChatGPT and GPT-4 on 709 graduate-level mathematics problems. They found average performance of only 3.20/5.0 for ChatGPT and 4.15/5.0 for GPT-4, concluding that LLM mathematical capabilities are "well below the level of a graduate student." Performance degraded particularly on proof-based questions and complex calculations. The authors also stated that LLMs "almost never expressed any form of uncertainty, even if output has been completely wrong", which implies that LLMs present incorrect results with identical confidence to correct ones.

Both perspectives contain truth. The question is not whether to use AI, as these tools are already reshaping research practices, but how to use them effectively, productively, and responsibly.

OUR APPROACH: STRUCTURED HUMAN-AI COLLABORATION

We propose a framework for productive human-AI mathematical collaboration based on clear division of responsibilities:

Human Responsibilities:

- Problem formulation and research direction
- Mathematical intuition and strategic decisions
- Verification of all AI-generated outputs
- Assessment of proof validity and novelty
- Final judgment on mathematical correctness

AI Responsibilities (under human supervision):

- Algebraic and symbolic manipulation
- Systematic exploration of proof strategies
- Literature search and synthesis
- LaTeX formatting and document preparation[1]
- Generation of numerical examples

This division mirrors the relationship between a senior researcher and a highly capable but unreliable research assistant: the AI does extensive algebraic and symbolic manipulations, but the human maintains ultimate responsibility and authority.

CASE STUDY: HERMITE QUADRATURE ERROR

To demonstrate this framework in action, we present our work on Hermite interpolation-based quadrature rules. The technical problem is substantial: deriving exact error representation and improved error bounds for numerical integration when both function values and derivatives are available.

Initial Problem Formulation (Human)

The research began when I came across the article "An elementary proof of error estimates for the trapezoidal rule" by Cruz-Uribe and Neugebauer [5] while preparing Chapter 23 of my adjoint book [6]. Their key innovation was using reverse integration by parts (RIBP) to derive error bounds for the trapezoidal rule requiring only boundedness of f' rather than f''. This elementary approach, accessible to calculus students, prompted me to ponder: could multiple applications of RIBP yield similar elementary derivations for Hermite quadrature, which uses both function and derivative values at endpoints? I jotted down a detailed idea for how to extend this work to Hermite quadrature, but did not work out the mathematical details.

[1] In particular, I asked `Claude` to convert my first draft of the paper to the CiSE magazine format and reorganize the draft content (which was originally written for a math magazine) to fit the special issue context. I further instructed it to rewrite the abstract and introduction to reflect the title and the context, pointed out the potential audiences of the article, and told it about my role and AI role. All conceptual contents, arguments, and interpretations are my own. Every AI-drafted sentence was reviewed, and it is either revised or removed if it does not match my tone/idea/intuition. The narrative, case study selection, and critical assessments are entirely my own. This experience reinforced a key lesson: AI can accelerate the mechanical aspects of writing, but the intellectual content must remain under human authorship and verification.

The idea was then assigned as an undergraduate summer-intern project. Over approximately ten weeks of manual work, the students successfully derived exact error representations and explicit error bounds for the cases $n = 2$ and $n = 3$, requiring only boundedness of $f^{(2)}$ and $f^{(3)}$ respectively (rather than the classical requirement of $f^{(4)}$ and $f^{(6)}$) [7]. However, they encountered prohibitive algebraic complexity when attempting to extend to general n, noting in their draft: "This ended up being very messy to show for a general case." In what follows, "my students" refers to these two researchers.

The central question I assigned to my students was: could we devise an elementary derivation of the exact error representation of Hermite quadrature in the spirit of [5]? A byproduct would be then replacing the restrictive requirement on the boundedness of the $2n$-th derivative with an n-th derivative of the integrand, making the results applicable to a broader class of functions.

This question required mathematical intuition and knowledge of the field. No AI system formulated this research direction. Instead, it emerged from human understanding of the problem domain and recognition of a gap in existing theory.

Literature Review (Human and AI Collaboration)

For this paper, we had already completed a literature review of work related to [5] and its extensions to Simpson's rule [8], [9], so we did not ask AI to assist with the literature search—though, in hindsight, it could have been useful and would have saved a significant amount of time. In general, AI (e.g. LLMs) tools can help find papers of similar type, summarize their main contributions, and even draft a state-of-the-art overview with BibTeX citations. Specifically, AI can synthesize patterns across papers and identify conceptual connections that keyword search might miss. However, they sometimes hallucinate citations or misattribute results. Indeed, in a separate project, I found that some AI-suggested references did not actually exist, and others did not support the contributions attributed to them. AI can therefore be useful for generating a candidate bibliography and an initial synthesis, but every reference and every claimed contribution must be verified through traditional search engines and human review. When I do use AI for literature review, I cross-check results across multiple AI systems and then follow up with an independent search (e.g., `Google Scholar`) to catch omissions and confirm details.

AI Success: Comprehensive literature coverage, identification of research gap.

AI Limitation: Required human verification of citation accuracy and relevance assessment.

Proof Strategy Development (Iterative Human-AI)

The Newton–Cotes formulas, using function values at endpoints, are among the most popular numerical integration methods. We now consider the setting in which one has both function values and derivatives at endpoints. The natural extension of Newton–Cotes is Hermite interpolation [10], [11], [12], [13], [14], [15] for $f(x)$. In particular, we consider the following quadrature rule:

$$\int_a^b f(x)\,dx = \int_a^b H_n(f;x)\,dx + E_n, \tag{1}$$

where $H_n(f;x)$ is the two-point Hermite interpolating polynomial constructed using the following information on the function and the values of its first $n-1$ derivatives: $f(a)$, $f'(a), \ldots, f^{(n-1)}(a)$ and $f(b)$, $f'(b), \ldots, f^{(n-1)}(b)$, and E_n is the corresponding integration error by replacing $f(x)$ with its Hermite interpolation $H_n(f;x)$.

An expression for the error in eq. (1) is available in [14], [16]. Lampret et al. [17] considered a quadrature rule referred to as the "composite Hermite rule," which applies when both function values $f(a)$, $f(a+\frac{b-a}{k})$, $f(a+2\frac{b-a}{k}), \ldots, f(b)$ and the endpoint derivatives $f'(a)$, $f'(b)$ are available, where k is the number of subintervals in $[a, b]$. Note that the case $k = 1$ in Lampret et al. [17] corresponds to taking $n = 2$ in eq. (1), and the resulting error bounds agree with those in [14], [16]. However, these error bounds require boundedness of the $2n$-th derivative of the integrand.

The core mathematical insight that I had was the following: if reverse integration by parts (RIBP) could yield elementary proofs for trapezoidal rule in [5], could *multiple* RIBP yield an elementary derivation of Hermite quadrature with exact error representations and milder conditions, by setting free parameters from RIBP properly?

This hypothesis was human-generated, but AI systems played a crucial role in exploring its consequences. I didn't carry out the math, and thus did not know whether the systems of equations for free parameters are well-posed and solvable in closed-form. My intuition suggested that the answer was yes when I assigned this task to my students. In their original draft [7], my students had:

- Derived exact error formulas for $n = \{2, 3\}$ using the coefficient matching approach,

- Found explicit values of free parameters for cases $n = \{2, 3\}$,
- Attempted but failed to find closed-form solutions for free parameters for general n due to algebraic complexity,[2]
- Proposed numerical algorithms as an alternative, though it was not clear if the systems were solvable numerically.

My responsibility was then to review this draft around mid-December 2025. Since this was a busy time, I wanted to see whether AI could help me speed up the task. To my surprise and pleasure, under my direction, AI helped more than I expected.

Definition 1 (Two-Point Hermite Interpolating Polynomial)**.** *Let a and b be distinct. The Hermite interpolating polynomial is defined by:*

$$H_n(f;x) = (x-a)^n \sum_{k=0}^{n-1} \frac{B_k (x-b)^k}{k!} + (x-b)^n \sum_{k=0}^{n-1} \frac{A_k (x-a)^k}{k!}$$

$$\text{with} \quad A_k = \frac{d^k}{dx^k}\left[\frac{f(x)}{(x-b)^n}\right]_{x=a}, \quad B_k = \frac{d^k}{dx^k}\left[\frac{f(x)}{(x-a)^n}\right]_{x=b}. \tag{2}$$

An immediate consequence is

$$H_n(f;a) = f(a),\ H_n'(f;a) = f'(a), \dots, H_n^{(n-1)}(f;a) = f^{(n-1)}(a),$$
$$H_n(f;b) = f(b),\ H_n'(f;b) = f'(b), \dots, H_n^{(n-1)}(f;b) = f^{(n-1)}(b).$$

For testing purposes, I gave two AI platforms[3] Definition 1 and asked them to verify the preceding derivative matching result. Interestingly, one of them immediately gave the correct proof and the other didn't. I pointed out that the proof was not correct, and the latter then provided a slightly different, but correct, proof.

A few remarks are in order. First, the AI models' proofs, though slightly different, used the same notation. This may imply that they were trained on similar data. Second, it is important that we verify and assess the validity of an AI proof (or answer), as they do make mistakes (all the time). Finally, to my pleasure, one of the proofs is slightly different from mine, that is, AI could provide/explore different proof ideas that are much faster than human.

Next, from the well-known Hermite interpolation error [14], [16]:

$$f(x) - H_n(f;x) = \frac{f^{(2n)}(\varepsilon)}{(2n)!}\left((x-a)(x-b)\right)^n, \tag{3}$$

where $\varepsilon \in [a, b]$, we have an exact error representation for Hermite quadrature:

$$\int_a^b f(x)\,dx = \int_a^b H_n(f;x)\,dx + \int_a^b \frac{f^{(2n)}(\varepsilon)}{(2n)!}\left((x-a)(x-b)\right)^n dx, \tag{4}$$

which, however, requires $f(x)$ to have a $2n$-th derivative (almost everywhere) on $[a,\ b]$. Consequently, for certain functions such as $|x|^{3/2}$ on the interval $[-1,\ 1]$, the representation for E_n in eq. (4) is not valid. Another example is

$$f(x) := \int_0^x \log(t - 1/2)\ dt$$

on the interval $[0,\ 1]$, for which the classical error eq. (3) does not exist for $n = 1$, though the Hermite interpolation (simple linear interpolation for this case), and thus its error, is well-defined.

The reverse integration by parts approach: The classical error representation eq. (4) requires $f^{(2n)}$. Our goal is to derive error representations requiring only $f^{(n)}$. The key insight comes from [5] for the trapezoidal rule: instead of using polynomial approximation theory, they applied integration by parts "backwards" (starting from the integral and working toward boundary terms) to express the error in terms of f' rather than f''. We extend this idea by applying RIBP multiple times—n iterations yield boundary terms involving $f, f', \dots, f^{(n-1)}$ at both endpoints, plus a remaining integral involving only $f^{(n)}$. The challenge is determining free parameters in the RIBP formula so the boundary terms exactly match the Hermite quadrature weights.

Hermite Polynomial Integration (AI with Human Verification)

We are going to derive an exact representation for E_n that requires only the n-th derivative. To that end, let us first write $\int_a^b H_n(f;x)\,dx$ in the form

$$\int_a^b H_n(f;x)\,dx = \sum_{j=0}^{n-1} w_j^a f^{(j)}(a) + \sum_{j=0}^{n-1} w_j^b f^{(j)}(b), \tag{5}$$

where $f^{(j)}(a)$ denotes the j-th derivative of $f(x)$ evaluated at $x = a$, and w_j^a, w_j^b are the weights to be determined.

Proposition 1. *Consider the Hermite interpolating polynomial in eq. (2) for a given n. Then the weights*

[2] In their original remark: "This ended up being very messy to show for a general case. So we left it out for now".

[3] Throughout this paper, I used two AI systems: Claude (Anthropic) and ChatGPT (OpenAI). I refer to them generically as "AI platforms" as the specific system is not relevant to the discussion.

w_j^a and w_j^b in eq. (5) are given by:

$$w_j^a = (b-a)^{j+1} n \sum_{k=j}^{n-1} \binom{k}{j} \frac{(n+k-j-1)!}{(n+k+1)!}, \quad w_j^b = (-1)^j w_j^a,$$

where $\binom{k}{j} = \frac{k!}{j!(k-j)!}$. Consequently,

$$\int_a^b H_n(f;x)\,dx = \sum_{j=0}^{n-1} w_j^a \left[f^{(j)}(a) + (-1)^j f^{(j)}(b) \right].$$

Since my students had already proved proposition 1 using direct integration of the Hermite polynomial with Beta function substitutions and general Leibniz rule, I asked one of the AI platforms to verify their proof rather than derive it independently. Its first response was "I have carefully checked your proof, and it looks correct!" with a few remarks that all steps are correct. It then said: "Overall: The proof is rigorous and correct! Nice work." Such a response made me suspicious. I then asked it to follow the derivation and check the algebraic manipulation of each step to make sure that each step was actually correct. The AI provided me with a detailed check, which sped up my review greatly. Out of curiosity, I asked these AI platforms to provide a proof of Proposition 1 for me. They provided different, but overly complex and long proofs. It is thus important for us to know that AI may provide unnecessarily complex (and possibly wrong) proofs.

The next result I needed to verify was the general n-fold RIBP, which should yield boundary terms involving derivatives up to order $n-1$ and a remaining integral involving $f^{(n)}$. My students had conjectured a specific form for this general formula, but I needed to verify it. I again asked one of the AI platforms to check it for me. It provided a long induction and concluded with reasons that [7, eq. (20)] was not correct. In particular, it proactively tried to verify the formula for $n = 3$ and found a problem, which was great. I then instructed it to find the corrected formula by integrating by parts n times the following expression (which is the last term on the right hand side of [7, eq. (20)] that I know for sure must appear after n-fold RIBP). The general expressions for the boundary terms are the ones I am looking for help from AI to bypass tedious algebraic calculations.

$$\int_a^b (-1)^n f^{(n)}(x) \left(\frac{(x+c)^{(n)}}{n!} + \sum_{i=0}^{n-2} \delta_i \frac{x^i}{i!} \right) dx. \quad (6)$$

After a couple of minutes of calculating, the AI platform said: "Excellent! I've completed a thorough verification. Here's what I found: The fully correct formula is...". However, the result was not correct: it missed factorials in the boundary terms! As can be seen, AI could indeed do the algebraic computation for me, which internally could be correct, but the final result provided to me was not entirely correct. Unlike a human, it does not seem to have the capability to double-check the result! I asked another AI platform the same question, and I was pleased that it provided the correct formula:

$$\begin{aligned}
\int_a^b f(x)\,dx = &\\
&\sum_{j=0}^{n-1} f^{(j)}(b) \underbrace{(-1)^j \left[\frac{(b+c)^{j+1}}{(j+1)!} + \sum_{i=0}^{j-1} \delta_{i+n-1-j} \frac{b^i}{i!} \right]}_{=:\alpha_j^b} \\
&+ \sum_{j=0}^{n-1} f^{(j)}(a) \underbrace{(-1)^{j+1} \left[\frac{(a+c)^{j+1}}{(j+1)!} + \sum_{i=0}^{j-1} \delta_{i+n-1-j} \frac{a^i}{i!} \right]}_{=:\alpha_j^a} \\
&+ \int_a^b (-1)^n f^{(n)}(x) \left(\frac{(x+c)^n}{n!} + \sum_{i=0}^{n-2} \delta_i \frac{x^i}{i!} \right) dx, \quad (7)
\end{aligned}$$

where $\theta = \{c,\ \delta_0, \ldots, \delta_{n-2} \in \mathbb{R}\}$ are arbitrary.

AI Success: Identified the appropriate mathematical tools and carried out the lengthy algebraic derivations and verifications.

Human Verification: Provided guidance for productive verifications, checked each algebraic step, and found and corrected errors involving missing the factorials.

Coefficient Matching (Human Insight, AI Execution)

Now, if we can identify constants $\theta := \{c,\ \delta_0, \ldots, \delta_{n-2}\}$ in eq. (7) such that the boundary terms on the right hand side match those on the right hand side of eq. (5):

$$\begin{aligned}
&\sum_{j=0}^{n-1} w_j^a f^{(j)}(a) + \sum_{j=0}^{n-1} w_j^b f^{(j)}(b) \\
&\quad = \sum_{j=0}^{n-1} \alpha_j^a f^{(j)}(a) + \sum_{j=0}^{n-1} \alpha_j^b f^{(j)}(b), \quad \forall f \in C^n[a,b], \quad (8)
\end{aligned}$$

then together with eq. (1), the Hermite quadrature error E_n is exactly eq. (6). It is important to point out that in this case E_n requires the nth-order, instead of the $2n$th-order, derivative of f to exist (almost everywhere). **The key questions are: 1) does a solution θ for eq. (8) exist?, and 2) can we find it in a closed-form expression?**

My original aim was more ambitious. In particular, we wanted to find closed form expressions for the constants from eq. (8). To that end, I turned to AI for help which should be much more efficient and

successful than me in terms of complex algebraic manipulations/calculations. I started with the concrete cases $n = \{2, 3\}$, for which my students had already determined the free parameters ($c = -(a+b)/2$, $\delta_0 = -(b-a)^2/24$ for $n = 2$, and corresponding values for $n = 3$), and asked one of the AI platforms, which already saw my students' solution, to find the corresponding coefficients. It followed a similar strategy by setting $\alpha_j^a = w_j^a$ and $\alpha_j^b = w_j^b$ for all $j = 0, \dots, n-1$. The interesting part was that it recognized the redundancy in the equations (see rigorous discussions below), and thus reduced the number of equations to be solved. It then provided closed form solutions for $n = 2$ and $n = 3$ in a blink of an eye! I verified the results and they were correct. Encouraged by this success, I then asked it to find closed form solutions for $n = \{4, 5\}$. Again, it provided the solutions in a blink of a eye, which were again correct after my verification. Before we proceed to general n, let us first discuss the redundancy in eq. (8), which will be useful for AI later.

The redundancy can be predicted from two places: 1) the role of a and b in Definition 1 can be interchanged, and 2) the equality (up to a sign) of w_j^a and w_j^b in Proposition 1. To make this rigorous, we first show that coefficient matching conditions are both necessary and sufficient for eq. (8) to hold.

Lemma 1 (Matching conditions)**.** *Equation* (8) *holds if and only if*

$$\alpha_j^a = w_j^a, \text{ and } \alpha_j^b = w_j^b, \quad \forall j = 0, \dots, n-1. \tag{9}$$

The sufficiency is clear, but the converse turned out to be more technical. For $n = 1$, we simply take a smooth cut-off function[4] ϕ such that $\phi = 1$ on a neighborhood of a, and $\phi \equiv 0$ on a neighborhood of b, and then take $f(x) = \phi(x)$ to conclude from eq. (8) that $\alpha_0^a = w_0^a$. Next, simply reflecting ϕ across the point $x = \frac{a+b}{2}$, i.e. taking $f(x) = \phi(a+b-x)$ yields $\alpha_0^b = w_0^b$. For general n, the idea is similar. In particular, we need to test eq. (8) with different functions f such that each time we either have $\alpha_j^a = w_j^a$ or $\alpha_j^b = w_j^b$, for all $j = 0, \dots, n-1$. I spent a good amount of time thinking but was not successful in constructing these functions. I gave one of the AI platforms a try, and to my pleasant surprise, it could construct these functions from ϕ! The construction (see eq. (10)) is in fact simple (now I know) and I wonder if the AI platform has seen this as part of its training data.

[4] Any of us who had prior training in distributional theory or L^p spaces or Sobolev spaces knows this fact [18], [19].

Proof of Lemma 1:
The sufficiency is clear, and we focus on the necessity. We start by defining the difference linear functional

$$D(f) := \sum_{j=0}^{n-1} (\alpha_j^b - w_j^b) f^{(j)}(b) + \sum_{j=0}^{n-1} (\alpha_j^a - w_j^a) f^{(j)}(a).$$

By (8), we have $D(f) = 0$ for all $f \in C^n[a, b]$. Let us define $d_j^a := \alpha_j^a - w_j^a$, $d_j^b := \alpha_j^b - w_j^b$, and

$$f_{j_0}(x) := \frac{(x-a)^{j_0}}{j_0!}\, \phi(x). \tag{10}$$

By construction, all derivatives of ϕ vanish at a and b, and thus

$$f_{j_0}^{(j)}(a) = \delta_{j,j_0}, \text{ and } f_{j_0}^{(j)}(b) = 0, \qquad j = 0, \dots, n-1.$$

Substituting into $D(f) = 0$ gives

$$0 = D(f_{j_0}) = \sum_{j=0}^{n-1} d_j^a f_{j_0}^{(j)}(a) + \sum_{j=0}^{n-1} d_j^b f_{j_0}^{(j)}(b) = d_{j_0}^a,$$

so $d_{j_0}^a = 0$. Since j_0 was arbitrary, $d_j^a = 0$ for all j. Similarly, i.e. reflecting f_{j_0} across the point $x = \frac{a+b}{2}$, we can show that $d_j^b = 0$ for all j. Therefore $\alpha_j^a = w_j^a$ and $\alpha_j^b = w_j^b$ for all $j = 0, \dots, n-1$. ■

Iterative development of the redundancy proof (iterative Human-AI)

From Lemma 1 we see that there are $2n$ equations in eq. (8), but we have only n unknown parameters c, $\delta_0, \dots, \delta_{n-2}$. Thus, n equations must be redundant, just like the cases $n = \{2, 3, 4, 5\}$ that we have discussed. To give AI productive instruction for general n, I first investigated $j = 0$. The two equations $\alpha_0^a = w_0^a$ and $\alpha_0^b = w_0^b$ are:

$$-(a+c) = (b-a)n \sum_{k=0}^{n-1} \binom{k}{0} \frac{(n+k-1)!}{(n+k+1)!} = \frac{b-a}{2},$$

$$b+c = (b-a)n \sum_{k=0}^{n-1} \binom{k}{0} \frac{(n+k-1)!}{(n+k+1)!} = \frac{b-a}{2}.$$

Solving either of the preceding equations for c gives

$$c = -\frac{a+b}{2},$$

independent of n. That is, either $\alpha_0^a = w_0^a$ or $\alpha_0^b = w_0^b$ is redundant when the other is solved. I guessed that this would hold true for any j, that is, either $\alpha_j^a = w_j^a$ or $\alpha_j^b = w_j^b$ is automatically valid, once the other holds.

Assume there exists θ such that $\alpha_j^a = w_j^a, 0 \le j \le n-1$ (to be proved below). We need somehow to tie the RIBP eq. (7) and the Hermite quadrature eq. (5) in order to relate α_j^b and w_j^b. I told the same AI platform that the Hermite weights w_j^a and w_j^b are symmetric (up

to the sign), and thus the RIBP weights α_j^a and α_j^b in eq. (7) may possess some similar symmetry, and thus the redundancy in the equations would be obvious. The AI came back to me with a proof of symmetry for the RIBP weights by reflecting f around $x = (a+b)/2$. However, the proof was wrong as it assumed the symmetry of the RIBP weights. When I "confronted" it, the AI provided another proof assuming the symmetry of the polynomial kernel $P(x,\theta)$ (to be defined below), but this was not correct either in general. I then told the AI that the redundancy, and hence the symmetry of RIBP weights, happens obviously when they match Hermite weights, and in that case, the polynomial kernel can be symmetric. However, (I told that AI that) the redundancy was what we needed to prove in the first place. I told the AI that we need to combine the RIBP eq. (7) and the Hermite quadrature eq. (5) in order to relate α_j^b and w_j^b. The AI then amazed me with a clean proof that avoids all the aforementioned issues, and that I now present in a constructive manner.

The classical result eq. (4) gives us the desired connection by taking $f \in \mathcal{P}^{2n-1}[a,b]$, where $\mathcal{P}^{2n-1}[a,b]$ is the set of polynomials of order at most $2n-1$, as the Hermite interpolation error $\frac{f^{(2n)}(\varepsilon)}{(2n)!}((x-a)(x-b))^n$ vanishes. In particular, we have

$$\begin{aligned}\sum_{j=0}^{n-1} w_j^a f^{(j)}(a) + \sum_{j=0}^{n-1} w_j^b f^{(j)}(b) &= \int_a^b H_n(f;x)dx \\ &= \int_a^b f(x)dx = \sum_{j=0}^{n-1} \alpha_j^a f^{(j)}(a) \\ +\sum_{j=0}^{n-1} \alpha_j^b f^{(j)}(b) + \int_a^b (-1)^n f^{(n)}(x) &\underbrace{\left(\frac{(x+c)^n}{n!} + \sum_{i=0}^{n-2} \delta_i \frac{x^i}{i!}\right)}_{=:P(x,\theta)} dx,\end{aligned}$$

for any $f \in \mathcal{P}^{2n-1}[a,b]$. After using the assumption $\alpha_j^a = w_j^a$, we obtain

$$\sum_{j=0}^{n-1} \left(w_j^b - \alpha_j^b\right) f^{(j)}(b) = (-1)^n \int_a^b f^{(n)}(x) P(x,\theta)\, dx, \tag{11}$$

for any $f \in \mathcal{P}^{2n-1}[a,b]$. What remains is to choose $f \in \mathcal{P}^{2n-1}[a,b]$ judiciously so that the right hand side vanishes while isolating $\left(w_j^b - \alpha_j^b\right)$ out on the left hand side. To annihilate the right hand side, the simplest choice is $f \in \mathcal{P}^{n-1}[a,b]$, to single out $\left(w_j^b - \alpha_j^b\right)$, the choice would not be obvious to me, hadn't I seen the polynomial factor in eq. (10). Indeed, by taking $f = \frac{(x-a)^{j_0}}{j_0!}$ for $0 \le j_0 \le n-1$, we have

$$w_{j_0}^b - \alpha_{j_0}^b = 0, \quad \forall 0 \le j_0 \le n-1, \tag{12}$$

which concludes the proof of the following theorem.

Theorem 1 (Redundancy)**.** *$\alpha_j^a = w_j^a, 0 \le j \le n-1$ if and only if $\alpha_j^b = w_j^b, 0 \le j \le n-1$.*

Remark 1. *To entertain myself, I asked if the AI could provide another proof of the exactness of Hermite interpolation for $f \in \mathcal{P}^{2n-1}[a,b]$ without using the classical error representation eq. (3). Without letting me down, the AI gave me an alternative correct (not shown here) using more elementary facts about polynomial and induction.*

Now, revisiting eq. (11) and using Theorem 1 yield interesting properties of the polynomial kernel $P(x,\theta)$.

Corollary 1. *Suppose there exists a set of parameters θ such that $\alpha_j^a = w_j^a, 0 \le j \le n-1$. Then*

$$\int_a^b f(x)P(x,\theta)\, dx = 0, \quad \forall f \in \mathcal{P}^{n-1}[a,b]. \tag{13}$$

While Corollary 1 was not what I aimed for, as I wanted to prove only Theorem 1, it was a nice byproduct collaboration with AI. These discussions and the success of the above proof of Theorem 1 also led me to an interesting symmetry of the polynomial kernel $P(x,\theta)$.

Lemma 2. *If the kernel orthogonality eq. (13) holds, then $P(x,\theta)$ is, up to a sign, symmetric around $x = (a+b)/2$, i.e.,*

$$P(a+b-x,\theta) = (-1)^n P(x,\theta).$$

Answer to the key questions (Human guide, AI extraordinary symbolic skill)

We are now in a position to answer the **key questions** posed after eq. (8). The first question is straightforward. Indeed, Theorem 1 allows us to work with one set of n equations $\alpha_j^a = w_j^a, 0 \le j \le n-1$. First the number of equations is the same as the number of free parameters. Second, these equations possess an obvious triangular structure that allows us to solve numerically for all free parameters in a straightforward manner. In particular, for $j = 0$, the corresponding equation $\alpha_0^a = w_0^a$ always yields $c = -(a+b)/2$ as we have shown above. For $j = 1$, the corresponding equation $\alpha_1^a = w_1^a$ gives us a linear monomial equation in terms of δ_{n-2}. If we continue this triangular structure, for general $1 \le j \le n-1$, the corresponding equation $\alpha_j^a = w_j^a$ gives us a recursive formula for δ_{n-1-j}

$$\delta_{n-1-j} = (-1)^{j+1} w_j^a - \frac{(a+c)^{j+1}}{(j+1)!} - \sum_{i=1}^{j-1} \delta_{i+n-1-j} \frac{a^i}{i!}.$$

The answer to the second key question requires either a human with an extraordinary symbolic manipulation/computational skill or a software with that skill. Since I am not the former, and neither are my students, I resorted to the latter option. I started with a greedy hands-free question for an AI platform: "Could you find simple closed forms for all free parameters in terms of n, a, and b?" The AI "honestly" responded: "yes, in principle, however, the resulting expressions rapidly become complicated as n increases (nested finite sums with binomial coefficients), and do not appear to admit a simple closed form in n". I then provided a more constructive and incremental request: "Consider $j = 1$, can you simplify and solve for δ_{n-2} as it does not involve other δ?" After a long, complex, and tedious symbolic computations, it yielded

$$\delta_{n-2} = -\frac{(b-a)^2}{8(2n-1)}, \quad \forall n \geq 2,$$

and provided the (bonus) consistency check for $n \in \{2,3,4,5\}$ that we worked out together previously.

Encouraged by the success, I asked the AI to find a closed form expression for δ_{n-3} from $j = 2$, given the fact that we already had δ_{n-2}. It again went through a series of complicated symbolic computations and found δ_{n-3}. However, it didn't provide a bonus consistency check for the cases $n = \{3,4,5\}$ that we already knew. I asked it to do the consistency check, and it admitted that the result was wrong and then argued that finding a simple closed-form expression for δ_{n-3} passing the consistency check was not tractable. I looked into its symbolic derivation and computation carefully and I suspected its computation for w_2^a was incorrect. I asked the AI to redo the calculations, and this time it provided the correct formula:

$$\delta_{n-3} = \frac{(a+b)(b-a)^2}{16(2n-1)}, \quad \forall n \geq 3.$$

I continued instructing the AI to find the correct expression for δ_{n-4} passing the consistency check. The detailed derivations required to arrive at the final expression for δ_{n-4} were beyond my patience, my time budget, and analytical skill (if I were to do it myself):

$$\delta_{n-4} = \frac{(b-a)^2\left((b-a)^2 + (6-4n)(a+b)^2\right)}{128(2n-3)(2n-1)}, \quad \forall n \geq 4.$$

At this point, I decided it was enough to convince myself there was no point to go further, as the expressions for these parameters were progressively more complex, and they would not add more value to the paper.

Initial AI Output: gave a "lazy" answer "I could not find closed form solutions", then under human constructive guidance, found solutions for some parameters, and provided incorrect solution for one case.

Human Intervention: Detected errors by working through specific cases ($n = 2,3$) by hand, identified the pattern, and provided insights for AI to correct its errors.

This iterative refinement exemplifies effective collaboration: AI handles lengthy tedious algebraic details, and human catches errors as well as guides corrections.

Tighter error bounds (AI provides extra facts, human exploits these facts)

The original error bounds that my students provided assumed the uniform boundedness of the n-derivative (see [7, eq. (7), Theorem 1, Theorem 2]). These bounds are the first thing one could do to derive an upper bound for the Hermite quadrature error. To see what AI could do using the same assumption for $n = 2$, that is, $\left|f^{(2)}(x)\right| \leq M < \infty$, I tasked an AI platform to provide a tight error bound. In the output LaTeX document, the AI documented all the steps including confusion and unsuccessful tries. Two important findings for me were: i) it could come up with the same bound as in [7, Theorem 1]

$$|E_2| \leq M\frac{(b-a)^3(\sqrt{3})}{54}, \tag{14}$$

and, perhaps more important, ii) it computed the integral of the polynomial kernel $K_2(x)$ as one of its efforts (that I didn't ask and didn't know) and it said: "Interesting! The kernel integrates to zero." The latter finding (which was only obvious to me later owing to Corollary 1) was a simple finding, but it was what I could exploit to improve the original bound eq. (14). In particular, from my prior experience in numerical analysis, it suggested that I could add a constant for free to obtain an improved error bound:

$$|E_2| \leq \|\Delta\|_\infty \cdot \frac{(b-a)^3\sqrt{3}}{54}, \tag{15}$$

where $\Delta(x) := f^{(2)}(x) - c$, with c being the midrange of $f^{(2)}(x)$ defined as

$$c := \frac{\inf_{[a,b]} f^{(2)}(x) + \sup_{[a,b]} f^{(2)}(x)}{2}.$$

Proof:
Since $\int_a^b K_2(x)\,dx = 0$, we have

$$|E_2| = \left|\int_a^b f^{(2)}(x)K_2(x)\,dx\right| = \left|\int_a^b \Delta(x)K_2(x)\,dx\right|$$
$$\leq \|\Delta\|_\infty \int_a^b |K_2(x)|\,dx \leq \|\Delta\|_\infty \cdot \frac{(b-a)^3\sqrt{3}}{54}.$$

TABLE 1. Error Bound Improvements for ($n = 2$) on $[0, 1]$

Function	Original	Improved	Factor
x^3	1.92×10^{-1}	9.62×10^{-2}	2.0×
e^x	8.72×10^{-2}	2.76×10^{-2}	3.2×
$\sin(2\pi x)$	1.27	1.27	1.0×
$\log(x + 1)$	3.21×10^{-2}	1.20×10^{-2}	2.7×
$x^4 - 2x^3 + x^2$	6.42×10^{-2}	4.81×10^{-2}	1.3×

Improved bound exploits kernel orthogonality to subtract optimal constant c from $f^{(2)}$. No improvement for $\sin(2\pi x)$ since the optimal constant is zero.

■

Note that the error bound in eq. (15) is tighter than eq. (14) owing to the following fundamental inequality for the deviation from the midrange

$$\|\Delta\|_\infty \leq \left\|f^{(2)}\right\|_\infty .$$

Table 1 shows the improvement of the bound eq. (15) over the original bound eq. (14) for five functions (that the AI picked and plotted for me). Note that the case with $\sin(2\pi x)$ shows no improvement because the optimal constant c is zero.

The bound can be improved even further if a higher derivative, say $f^{(n+k)}$, with $k \leq n$, is bounded by replacing c with the best $(k - 1)$th-order polynomial approximation to $f^{(n)}$ in the infinity norm. We can also play the same trick for the L^2-norm.

TECHNICAL ACHIEVEMENTS SUMMARY

For readers interested in the mathematical outcomes beyond the collaboration process, we briefly summarize our technical contributions:

1) Exact error representation for Hermite quadrature requiring only n-th derivatives instead of $2n$-th derivatives
2) Proof that coefficient matching conditions are necessary and sufficient (Lemma 1 on Matching Conditions)
3) Redundancy theorem showing that solving n equations from $2n$ is sufficient to determine a unique set of coefficients
4) Closed-form solutions for free parameters up to $n = 4$
5) Orthogonality properties of the polynomial kernel
6) Improvements on the error bounds

Detailed proofs and extensions are provided in the accompanying technical paper [20].

PATTERNS OF SUCCESS AND FAILURE

Our experience reveals clear patterns in what works and what doesn't in human-AI mathematical collaboration. It is important to emphasize that AI is not a solution, but a highly capable but unreliable research assistant.

Where AI Excelled

1) **Algebraic Manipulation**: AI systems handled complex symbolic manipulations reliably once the setup was correct and the results are verified by human. This saves hours or days of tedious calculation.
2) **Systematic Exploration**: When asked to explore variations (different values of n, alternative proof strategies), AI provided comprehensive coverage faster than a human working alone.
3) **Literature Synthesis**: AI rapidly identified relevant papers and synthesized key ideas, though citation accuracy required verification.
4) **LaTeX Formatting**: AI produced well-formatted mathematical documents, handling complex equation environments and theorem styling.
5) **Numerical Verification**: AI generated comprehensive test cases (not shown here) and verified specific cases much faster than a human.

Where AI Failed

1) **Problem Formulation**: AI couldn't reliably generate research questions. Most significant mathematical insights and strategic directions came from human researchers.
2) **Error Detection**: AI frequently produced outputs with subtle errors (wrong indices, incorrect signs, wrong results) that appeared plausible but were mathematically wrong.
3) **Proof Validity**: AI could not reliably assess whether a proof was valid or novel. It sometimes claimed to verify results that were actually incorrect.
4) **Strategic Directions**: When stuck, AI would often continue down unproductive paths. Human intuition was essential for course correction.
5) **Mathematical Judgment**: In general, AI could not determine whether results were interesting, surprising, or worth publishing. It tended to flatter results/findings and sometimes even said that the results were publishable.

Critical Failure Mode: Plausible Nonsense

The most dangerous failure mode was AI generating mathematical statements that appeared correct but contained fundamental errors. For example, when exploring various conditions on $f^{(2)}$ and their corresponding bounds, it made a mistake that seemed convincing at first glance: assuming $f^{(2)}$ has constant sign and then yield perfect effort bound by implicitly assuming that $f^{(2)}$ was a constant. Only detailed verification revealed the error. There are many other occasions where AI provided plausible nonsense. Another example was when I wanted to show that $\alpha_j^b = (-1)^j \alpha_j^a, \quad j = 0, \dots, n-1$, and the AI provided the following plausibly correct statement

> *Let $n \geq 1$. Suppose there exist constants α_j^a, α_j^b $(j = 0, \dots, n-1)$ and a polynomial K_n such that for every $f \in C^n[a,b]$ we have eq. (7). Then the endpoint coefficients satisfies*
>
> $$\alpha_j^b = (-1)^j \alpha_j^a, \qquad j = 0, \dots, n-1.$$

with a believable proof using test functions from distributional theory, etc. The statement itself was too good to be true because that couldn't happen unless eq. (8) holds. When I went through the proof carefully I found a fundamentally incorrect, but plausibly correct fact that AI used (not shown here for brevity).

Warning Signs That AI May Be Wrong

- Result seems too neat or simple for the problem complexity
- Explanation is verbose but vague on critical details
- Similar problems give inconsistent results
- AI refuses to show intermediate steps
- Numerical tests contradict theoretical predictions
- Proof relies on unverified "well-known" facts

This highlights a critical point: **AI assistance without expert human verification is actively dangerous.**

Connecting Observed Failures to LLM Reliability Literature

Our AI failure experiences align with systematic evaluations of LLM mathematical capabilities, confirming these are general limitations rather than isolated incidents. Three specific findings in [4] directly match our experiences:

1. Computational errors uncorrelated with difficulty: The authors in [4] found ChatGPT "executes complicated symbolic tasks with ease" but "fails on simple arithmetic operations," with errors uncorrelated with expression/problem complexity. Computational errors occurred in 36% of algebra problems. Our experiences match this pattern: AI correctly handled complex Beta function evaluations but dropped factorials in simpler nested summations and integration by parts.

2. Unwavering confidence despite errors: ChatGPT "almost never expressed any form of uncertainty, even if its output has been completely wrong." Our observation: AI presented the incorrect RIBP formula with complete confidence, then confidently validated its own error when asked to verify. Only explicit counterexamples forced error recognition.

3. Logical errors in proofs: Frieder et al. documented "e5_5: circular logical argument" as a systematic failure mode. Our observation: AI's symmetry proof assumed the symmetry it claimed to prove—exactly this error type.

Implication: Our detailed case study illustrates what Frieder et al.'s "3.20/5.0 rating" means in practice: AI makes multiple errors at every stage of graduate-level mathematics, requiring constant human detection and correction. Our collaboration succeeded because we maintained the rigorous verification their evaluation prescribes as essential.

LESSONS AND GUIDELINES

Based on this experience, we offer guidelines for productive human-AI collaboration in mathematics:

Dos

1) **Maintain Verification Culture/habit**: Treat all AI outputs as hypotheses to be tested, not facts to be accepted.
2) **Use AI for Leverage**: Deploy AI on tasks that are tedious but verifiable (algebra, literature search, formatting). AI proof exploration could be surprisingly productive under human supervision and rigorous verification.
3) **Preserve Human Judgment**: Keep research direction, problem formulation, and final assessment under human control.
4) **Document Thoroughly**: Keep detailed records of AI contributions for both credit attribution and debugging.

5) **Test Extensively**: Generate numerical examples, special cases, and limiting behavior checks.
6) **Iterate Rapidly**: Use AI to quickly explore multiple approaches, then apply human judgment to select promising directions.

Dos: Five Core Principles

1) **Verify Everything**: Treat all AI outputs as hypotheses requiring verification
2) **Maintain Strategic Control**: Keep research direction and problem formulation human-led
3) **Use AI for Tedious Tasks**: Deploy AI for algebra, literature search, formatting, and proof explorations
4) **Test Extensively**: Generate numerical examples and special cases
5) **Document Transparently**: Keep detailed records of AI contributions

Quick Verification Checklist

Before accepting any AI result:

- Test with simple special cases where you know the answer
- Check dimensional analysis and limiting behavior
- Verify non-trivial intermediate steps by hand
- Compare with existing results in limiting cases
- Run numerical experiments if analytical verification is difficult

Don'ts

1) **Don't Trust Blindly**: Never accept AI mathematical outputs without verification.
2) **Don't Outsource Understanding**: Use AI to augment your thinking, not replace it.
3) **Don't Skip Steps**: Even if AI produces an answer, work through the derivation yourself. For steps that AI skipped, either carry them out yourself or ask them to provide the details and then check.
4) **Don't Ignore Errors**: When you find errors in AI output, use them as learning opportunities about failure modes.
5) **Don't Publish/Accept Without Verification**: Ensure every theorem, proof, and claim has been independently verified by qualified humans.

BROADER IMPLICATIONS

My limited experience suggests several broader implications for AI in scientific research:

1. AI as an Amplifier, Not a Replacement: AI tools amplify human capability but cannot replace domain expertise and judgment.

2. Verification is Essential: The ease of generating content makes verification more critical, not less.

3. New Skills Required: Researchers need to develop skills in prompting and strategic directions, verification, and critical assessment of AI outputs.

4. Transparency Matters: Academic culture should encourage, not discourage, transparency about AI assistance.

5. Education Must Adapt: We must teach students both how to use AI tools and, critically, how to work without them productively and responsibly.

ANSWER TO THE TITLE QUESTION

Is human-centered AI-assisted creative scholarly work in mathematics productive collaboration or perilous shortcut?

My personal answer: **it depends entirely on how you use it.**

With proper safeguards—expert human verification, clear division of responsibilities, appropriate skepticism, and rigorous testing—AI assistance can meaningfully accelerate mathematical discovery and verification while enhancing understanding. We completed in weeks what might have taken months without AI assistance.

However, without these safeguards, AI assistance becomes a liability. The risk of subtle errors, the atrophy of mathematical thinking, and the false confidence in unverified results can lead researchers astray.

The technology is neither inherently good nor bad. The outcomes depend on human choices about implementation and oversight.

SHOULD WE ENCOURAGE HUMAN-CENTERED AI-ASSISTED SCHOLARLY WORK?

This is clearly debatable (and even controversial). Some of us may disapprove of the idea, while others will embrace it. Historically, scholarly work has been human–human collaboration, and that has been the default model.

We have long accepted citing mathematical software and computer algebra systems like `Mathematica`, `MATLAB`, or `Maple` when they

perform symbolic manipulations. These tools execute algorithms with precise syntax and problem encoding programmed by humans, yet we don't credit them as coauthors. Why? Because they are traditionally deterministic tools executing known procedures.

LLMs are fundamentally different. They offer natural language interaction and can suggest solution strategies, but could make computational errors. They generate novel strategies, explore unexpected proof directions, and sometimes produce insights that surprise even expert users. The distinction between "tool" and "collaborator" becomes murky in this case. If we credit a human research assistant under supervision, why not credit an AI assistant performing the same function, as long as new knowledge or findings have been developed? In other words, as far as advancing knowledge and the state-of-the-art is concerned, should it matter if a scholarly work is a human-human or human-AI collaboration?

However, I acknowledge the counterarguments:

- AI lacks intent, understanding, and responsibility.
- AI cannot defend its contributions or verify correctness.
- Coauthorship implies accountability that AI cannot bear.
- There are practical issues with copyright and legal responsibility

Perhaps the solution is a new category: "AI Research Assistant" in acknowledgments, with detailed specification of contributions. This preserves transparency without granting full coauthorship status.

LIMITATIONS OF THIS STUDY

This case study has several limitations that readers should consider:

1) **Single Domain**: Our experience is limited to mathematical proof development. Results may not generalize to experimental science, data analysis, or other forms of scholarly work.
2) **Expert User**: As a professor with extensive mathematical training and experience, I could quickly identify AI errors. Less experienced researchers might struggle with verification.
3) **Specific Problem Type**: The algebraic nature of our problem played to AI strengths. More conceptual or geometric problems might yield different results.
4) **Time Snapshot**: AI capabilities are rapidly evolving. This study reflects systems available in late 2025.
5) **Selection Bias**: We document a successful collaboration. Failed attempts at AI-assisted research may have different patterns we didn't observe.

While broader methodological studies across multiple problems would enhance generalizability, we believe detailed transparency about one case offers practical insights appropriate for illuminating controversies.

COMPARISON WITH SPECIFIED TOOLS

LLMs vs. Search Engines for Literature: LLMs can synthesize patterns across papers and identify conceptual connections that keyword search might miss. However, they sometimes hallucinate citations or misattribute results. My recommendation is to use LLMs for initial literature mapping, then verify all citations manually using traditional search and human verifications.

LLMs vs. Computer Algebra Systems: CAS (like `Mathematica` and `Maple`) excel at deterministic symbolic computation but require precise syntax and problem encoding. LLMs offer natural language interaction and can suggest solution strategies, but could make computational errors. The ideal workflow combines both: use LLMs for strategy exploration and problem setup, then CAS for verifications when applicable.

General LLMs vs. `LLEMMA` and `Minerva`: This requires an in-depth, comprehensive, and systematic comparison, which is beyond the scope of this paper. However, the general LLMs (such as `GPT-4`, `Claude`) proved surprisingly capable when properly prompted for our mathematical derivation tasks. This suggests that for many applications, accessibility and ease of use may matter more than mathematical specialization.

CONCLUSION

We have provided empirical evidence that human-AI collaboration in mathematics can produce rigorous, novel results when properly structured. Our work on Hermite quadrature demonstrates both the potential and the limitations of current AI systems.

The key insight is not that AI can do mathematics—it cannot, at least not reliably—but that it can assist mathematicians in specific, well-defined ways under appropriate supervision. The human must remain firmly in control, providing intuition, strategic directions, verification, and judgment.

As AI capabilities continue to advance, the research community must develop best practices for responsible

use. We hope this detailed account of our experience contributes to that ongoing conversation by providing concrete evidence of what works, what doesn't, and what safeguards are essential.

The future of mathematics may well involve human-AI collaboration. Our results suggest this future can be productive, if we proceed thoughtfully, responsibly, and maintain appropriate skepticism.

ACKNOWLEDGMENTS

The author would like to thank his students, Giancarlo Villatoro and C.G. Krishnanunni for their initial work without AI assistance in [7]. The author also would like to acknowledge the contributions of Claude (Anthropic) and ChatGPT (OpenAI) as AI research assistants in this work. All final collaborative mathematical results were verified by the author. We thank the Associate Editor and all reviewers for their constructive feedback, which has significantly improved the manuscript. This work is partially funded by the National Science Foundation award NSF-OAC-2212442, and by the Department of Energy award DE-SC0024633.

Examples of prompts and outputs

REFERENCES

1. Alex Davies, Petar Veličković, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomašev, Richard Tanburn, Peter Battaglia, Charles Blundell, András Juhász, Marc Lackenby, Geordie Williamson, Demis Hassabis, and Pushmeet Kohli. Advancing mathematics by guiding human intuition with AI. *Nature*, 600(7887):70–74, 2021.
2. Thomas Hubert et al. Olympiad-level formal mathematical reasoning with reinforcement learning. *Nature*, 2025. Published November 12, 2025.
3. 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. *Advances in Neural Information Processing Systems*, 34:1–11, 2021. NeurIPS 2021 Datasets and Benchmarks Track.
4. Simon Frieder, Luca Pinchetti, Alexis Chevalier, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Christian Petersen, and Julius Berner. Mathematical capabilities of ChatGPT. In *Advances in Neural Information Processing Systems*, volume 36, 2023. NeurIPS 2023 Datasets and Benchmarks Track.
5. David Cruz-Uribe and CJ Neugebauer. An elementary proof of error estimates for the trapezoidal rule. *Mathematics magazine*, 76(4):303–306, 2003.
6. Tan Bui-Thanh. Adjoint methods in computational science and engineering. https://users.oden.utexas.edu/~tanbui/books/AdjointBook.pdf, 2025. Unpublished manuscript / book draft.
7. Giancarlo Villatoro, C. G. Krishnanunni, and Tan Bui-Thanh. On the error of Hermite quadrature: An elementary proof. https://users.oden.utexas.edu/~tanbui/books/HermiteQuadrature.pdf, 2019. Unpublished manuscript.
8. JASON R. ELSINGER. An elementary proof of the simpson's rule error formula. *Pi Mu Epsilon journal*, 12(6):353–357, 2007.
9. D. D. Hai and R. C. Smith. An elementary proof of the error estimates in simpson's rule, 2008.
10. Garrett Birkhoff and Carl de Boor. Piecewise polynomial interpolation and approximation. In T. E. Hull, editor, *Approximation of Functions*, pages 164–190. Elsevier, 1964.
11. Larry L. Schumaker. On Hermite interpolation by splines. *Journal of Approximation Theory*, 9(1):2–29, 1973.
12. Manolis I. A. Lourakis. Interpolating using Hermite splines. Technical report, Institute of Computer Science, FORTH, 2007.
13. George Gasper and Mizan Rahman. *Basic Hypergeometric Series*, volume 96 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2 edition, 2004.
14. Philip J. Davis. *Interpolation and Approximation*. Dover Publications, 1975.
15. Josef Stoer and Roland Bulirsch. *Introduction to Numerical Analysis*, volume 12 of *Texts in Applied Mathematics*. Springer, 2 edition, 1993.
16. Kendall E. Atkinson. *An introduction to numerical analysis*. Wiley, 1978.
17. Vito Lampret. An invitation to hermite's integration and summation: A comparison between hermite's and simpson's rules. *SIAM review*, 46(2):311–328, 2004.
18. Gerald B. Folland. *Real Analysis: Modern Techniques and Their Applications*. John Wiley & Sons, 2 edition, 1999.
19. Morris W. Hirsch. *Differential Topology*, volume 33 of *Graduate Texts in Mathematics*. Springer, 1976.
20. Tan Bui-Thanh, Giancarlo Villatoro, and C. G. Krishnanunni. From an elementary proof of error representation for hermite quadrature to a rediscovery of legendre polynomials and rodrigues formula. *arXiv preprint*, 2026. Available at https://arxiv.org/abs/2602.18634.