SENIOR SEMINAR IN MATHEMATICS

AI IN MATHEMATICS RESEARCH A COMPREHENSIVE GUIDE FOR SENIOR UNDERGRADUATES



R. L. HERMAN - VERSION DATE: JULY 16, 2025

Copyright © 2025 by Senior Seminar in Mathematics

PUBLISHED FOR R. L. HERMAN

This text has been formatted using a modification of the Tufte-book documentclass in $\[MT_EX]$. See tufte- $\[MT_EX]$.googlecode.com.

AI IN MATHEMATICS RESEARCH A COMPREHENSIVE GUIDE FOR SENIOR UNDERGRADUATES by Russell Herman is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License.

First printing, 2025

Contents

1	AI and Its Impact on Mathematics			
2 A Quick Start Workflow: From Probl		Vorkflow: From Problem to Proof	5	
	2.1 Step 1: Pro	blem Analysis and Strategy	5	
	2.2 Step 2: Coc	le Generation	6	
	2.3 Step 3: Ver	ification and Visualization	7	
	2.4 Step 4: Dis	semination	8	
3	Background and	cground and History of AI in Mathematics 11		
4	Current Tools a	nd Technologies	15	
	4.1 Large Lang	guage Models and Chatbots	15	
	4.2 Computer	Algebra Systems	16	
	4.3 AI Image a	nd Video Generation	19	
	4.4 Proof Assis	stants and Formal Systems	20	
5 A Mathematician's Guide to Ho		an's Guide to How LLMs Work	23	
5	5.1 Words as V	Vectors	23	
	5.2 Attention:	A Dynamic, Weighted Average	24	
	5.3 Prediction	as a Probability Problem	24	
6	Effective Promp	ective Prompting Strategies 27		
7	LLM Chaining	A Chaining		
8	AI-Assisted Ma	AI-Assisted Mathematical Problem Solving		
	8.1 Solving Or	dinary Differential Equations	36	
	8.2 Solving Par	rtial Differential Equations	37	
	8.3 Data Analy	vsis and Statistics	40	
	8.4 Combinato	rics and Graph Theory	41	
	8.5 Pure Mathe	ematics: Number Theory	43	
	8.6 Advanced	Prompting Techniques	44	
	8.7 Optimizati	on and Machine Learning	45	
	8.8 Capstone R	Research Projects	47	
	8.9 Conclusion	and Future Directions	48	
9	Using AI for Pre	oblem Solving	51	

10	Code Generation and Computation	55
11	Verification and Validation Results	61
12	Creating Graphics and Visualizations	65
	12.1 Integration of AI with Mathematical Visualization Tools	65
	12.2 The Art and Science of Mathematical Graphics	69
	12.3 Refining Graphics	72
13	LATEXDocument Preparation	75
14	Presentations and Beamer	81
15	Case Study: From Conjecture to Conclusion	89
	15.1 The Spark of Curiosity: Problem Formulation	89
	15.2 Computational Exploration: Code Generation	90
	15.3 Verification and Visualization	92
	15.4 Dissemination: $LATEX$ and Ethical Disclosure	93
16	Ethics and AI-Assisted Mathematics	95
	16.1 Intellectual Integrity and Attribution	96
	16.2 Transparency and Disclosure Requirements	97
	16.3 Verification and Critical Evaluation	98
	16.4 Bias and Fairness in AI-Assisted Research	99
	16.5 The Future of Mathematical Authorship	100
	16.6 Ethical Guidelines for Practice	101
17	Research Methodology and Best Practices	105
	17.1 Planning AI-Assisted Research Projects	105
	17.2 Collaborative Workflows and Documentation	107
	17.3 Quality Assurance and Peer Review	108
	17.4 Publication and Dissemination Strategies	109
18	Future Directions and Emerging Technologies	111
	18.1 Emerging AI Technologies for Mathematics	111
	18.2 Integration with Traditional Mathematical Tools	112
	18.3 Challenges and Limitations	113
	18.4 Implications for Mathematical Education	114
	18.5 Long-term Vision for Al-Assisted Mathematics	115
19	Conclusion and Recommendations	117
	19.1 Key Principles for Responsible AI Use	117
	19.2 Building AI Literacy for Mathematical Research	118
	19.3 Developing Effective Research Workflows	119
	19.4 Preparing for Future Developments	119
	19.5 Contributing to the Mathematical Community	120
	19.6 Final Thoughts	120

Α	Resources and Further Reading		
	A.1 AI Tools and Platforms	123	
	A.2 Educational Resources	124	
	A.3 Professional Organizations and Communities	124	
	A.4 Conferences and Workshops	125	
B	Bibliography		
C	Index	133	

1____AI and Its Impact on Mathematics

The question of whether a computer can think is no more interesting than the question of whether a submarine can swim.

— Edsger W. Dijkstra (1930 – 2022)

Mathematics is beautiful, and it should be presented beautifully. 数学是美的,应该美地呈现出来。

— Shiing-Shen Chern (陈省身)

THE INTEGRATION OF ARTIFICIAL INTELLIGENCE (AI) into mathematics research represents one of the most significant paradigm shifts in the field since the advent of computer algebra systems. As we stand at the threshold of a new era in mathematical discovery and communication, it becomes imperative for emerging mathematicians to understand not only how to leverage these powerful tools but also to appreciate their limitations, ethical implications, and the fundamental changes they bring to the research process.

Artificial intelligence, in its current manifestation through large language models (LLMs) and specialized mathematical AI systems, is fundamentally altering how we approach problem-solving, hypothesis generation, proof verification, and even the communication of mathematical ideas. This transformation extends beyond mere computational assistance; it touches the very core of how mathematical knowledge is created, verified, and disseminated within the academic community.

The practice of mathematics has always been a dialogue between human intuition and formal structure. The tools of this dialogue have evolved: from the sand-drawn diagrams of Euclid, to the chalk and blackboard of the 20th century, to the computational powerhouses of today. We are now at the threshold of another evolutionary leap, driven by the widespread availability of Artificial Intelligence. For the student of mathematics, these tools are not merely a novelty; they represent a fundamental change in how we can search for answers, build models, analyze data, and even conceive of new mathematical ideas.

The historical trajectory of AI development in mathematics can be traced back to the early symbolic computation systems of the 1960s, but the contemporary landscape is dominated by neural network architectures that can process natural language, generate mathematical content, and even assist in complex reasoning tasks. These systems represent a qualitative leap from traditional computer algebra systems, offering capabilities that blur the line between computational assistance and intellectual collaboration.

Understanding this technological evolution is crucial for several reasons. First, it enables mathematicians to make informed decisions about when

This guide is designed to be a comprehensive resource for navigating this new landscape. It aims to equip you with the practical skills and, more importantly, the critical mindset needed to use AI effectively and ethically. We will treat AI not as a replacement for your own intellect, but as a powerful amplifier for it-a tool for becoming an augmented mathematician.

and how to incorporate AI tools into their research workflows. Second, it provides the necessary context for evaluating the reliability and appropriateness of AI-generated content. Third, it helps researchers maintain the integrity of mathematical discourse while embracing innovation. Finally, it prepares the next generation of mathematicians to navigate an increasingly AI-integrated academic landscape.

The impact of AI on mathematical research extends across multiple dimensions. In problem-solving, AI systems can suggest novel approaches, identify patterns in large datasets, and even generate conjectures that might not be immediately obvious to human researchers. In proof verification, AI tools are becoming sophisticated enough to check the logical consistency of mathematical arguments, though they still require careful human oversight. In communication, AI is revolutionizing how mathematical papers are written, formatted, and presented, offering new possibilities for clarity and accessibility while raising important questions about authorship and intellectual contribution.

THIS GUIDE IS STRUCTURED TO TAKE YOU from foundational understanding through practical implementation to research-level application. The progression follows a natural learning sequence that mirrors how you would actually incorporate AI into your mathematical work.

Part I: Foundations (Chapters 1–5) establishes the conceptual groundwork. After this introduction, Chapter 2 provides an immediate hands-on experience with the complete AI-assisted research workflow—from problem formulation through verification to publication. Chapters 3–4 survey the historical context and current landscape of AI tools, helping you understand what's available and how to choose appropriate tools for different tasks. Chapter 5 explains how large language models work from a mathematician's perspective, providing the technical understanding necessary for effective use. These chapters answer the fundamental questions: *What is AI-assisted mathematics? What tools are available? How do they work?*

Part II: Core Research Skills (Chapters 6–8) develops the practical abilities you need for effective AI collaboration. Chapter 6 teaches prompting strategies—the art of communicating effectively with AI systems to get high-quality mathematical assistance. Chapter 7 introduces LLM chaining for complex, multi-step tasks that require sophisticated reasoning. Chapter 8, the heart of the book, demonstrates AI-assisted problem solving across diverse mathematical domains—from differential equations to number theory—through worked examples and guided exercises. These chapters answer: *How do I work effectively with AI? How do I solve mathematical problems using AI assistance?*

Part III: Implementation and Communication (Chapters 9–14) covers the technical and communication aspects of AI-assisted research. You'll learn code generation and verification (Chapters 9–11), graphics creation (Chapter 12), and document preparation with LATEX and Beamer (Chapters 13–14). These practical chapters ensure you can implement, test, and communicate your AI-assisted mathematical work professionally. These chapters answer: *How do I implement, verify, and communicate my AI-assisted work?*

Part IV: Research Practice (Chapters 15–18) addresses the professional and ethical dimensions of AI-assisted mathematics. A complete case study (Chapter 15) demonstrates the full research process from conjecture to publication, while Chapter 16 provides essential ethical guidelines for responsible AI use. Chapters 17–18 cover research methodology and future directions, preparing you for the evolving landscape of mathematical research. These chapters answer: *How do I conduct responsible AI-assisted research? What does the future hold?*

Students can follow the complete sequence for comprehensive preparation, or focus on specific sections based on immediate needs. Those wanting quick practical skills might start with Chapter 2, then proceed to Chapter 8. Students primarily interested in code generation could focus on Chapters 6, 10, and 11. The extensive cross-references throughout support flexible navigation while maintaining coherence.

This guide is designed to provide senior undergraduate mathematics students with a comprehensive understanding of how to thoughtfully and effectively integrate AI tools into their research practices. We will explore the technical capabilities and limitations of current AI systems, examine the ethical considerations that arise from their use, and provide practical guidance for maintaining scholarly integrity while leveraging these powerful technologies. At the same time we must keep in mind that this technology is changing so fast, that some of this guide could already be outdated since this was last revised July 16, 2025.

At the same time we must keep in mind that this technology is changing so fast, that some of this guide could already be outdated since this was last revised July 16, 2025.

Acknowledgments

THIS GUIDE WAS ORIGINALLY GENERATED through collaboration with Claude (Anthropic's AI assistant) on July 9, 2025, based on a comprehensive prompt for creating a mathematics research guide. The content represents AI-generated material that has been reviewed and refined through iterative dialogue. It has been reformatted and additional material added.

The image on the cover was generated using ChatGPT-40's image capabilities ¹. The prompt was

Prompt: Create an image for a textbook on using AI in mathematics research

This produced the image in Figure 1.1. It was then cropped to a more satisfactory cover image.

USING AI IN MATHEMATICS RESEARCH ¹ OpenAI. Image generated by ChatGPT-40 based on user prompt. ChatGPT, July 2025

Figure 1.1: The cover image was generated using ChatGPT-40 (via DALL·E).

2 A Quick Start Workflow: From Problem to Proof

The best way to have a good idea is to have a lot of ideas.

- Linus Pauling

In mathematics, the art of proposing a question must be held of higher value than solving it. 提出问题比解决问题更重要。

— Hua Luogeng (华罗庚)

BEFORE WE DELVE INTO THE THEORY AND SPECIFICS of each tool, let's walk through a complete, albeit simple, research workflow from start to finish. This chapter will serve as a practical demonstration of the core methodology you will see throughout this guide. Our goal is to take a standard undergraduate-level problem and solve it not just with an answer, but with analysis, code, verification, and a formal summary—all with an AI as our collaborator.

Our problem is a classic one from numerical analysis:

Find the real root of the polynomial $f(x) = x^5 - x - 1$.

This is a good first problem because it has no simple algebraic solution via radicals, as established by Abel-Ruffini theory. We are forced to use numerical and analytical methods. In this workflow, you are the mathematician and project director; the AI is your exceptionally fast and knowledgeable, but un-trusted, research assistant.

2.1 Step 1: Problem Analysis and Strategy

BEFORE WRITING ANY CODE, a good mathematician analyzes the problem. Does a root exist? Is it unique? What methods are suitable? We can use an AI to quickly survey the problem landscape.

Prompt to AI

Analyze the function $f(x) = x^5 - x - 1$. I need to find its real roots. 1. Prove that at least one real root exists. 2. Determine how many real roots there are. 3. Suggest a suitable numerical method for finding the root. This first step is crucial. We use the AI not for a final answer, but for reconnaissance and strategy. It helps us confirm our intuition and choose a path forward.

Start with problem.

Al Output

This is a great problem that demonstrates the power of calculus and numerical methods.

1. **Existence of a Root:** We can use the Intermediate Value Theorem (IVT).

- $f(1) = 1^5 1 1 = -1$
- $f(2) = 2^5 2 1 = 29$

Since f(x) is a polynomial, it is continuous everywhere. Because f(1) < 0 and f(2) > 0, the IVT guarantees there is at least one root $c \in (1, 2)$.

2. **Number of Real Roots:** To determine the number of real roots, we analyze the derivative:

$$f'(x) = 5x^4 - 1$$

The critical points occur where f'(x) = 0, which is $x^4 = 1/5$, so $x = \pm (1/5)^{1/4}$. These are the locations of the local extrema.

•
$$f((1/5)^{1/4}) \approx -1.32$$

•
$$f(-(1/5)^{1/4}) \approx -0.68$$

Since both local extrema are negative, the function crosses the x-axis only once. Therefore, there is exactly one real root.

3. **Suggested Method:** Given that we have a good analytic understanding of the function, Newton's method would be very efficient. However, a more robust and simpler approach for a one-off problem is to use a library function like a bisection or secant solver, such as scipy.optimize.root_scalar in Python.

The AI's analysis confirms the existence of a unique real root and suggests a robust path forward: using a pre-built numerical solver from a trusted library.

2.2 Step 2: Code Generation

WITH A CLEAR STRATEGY, we can now ask the AI to implement it. We will ask for a Python script, as its scientific libraries are perfectly suited for this task.

Prompt to Al

Write a Python script using the SciPy library to find the unique real root of $f(x) = x^5 - x - 1$.

Here, the AI acts as a code generator. It translates our English-language strategy into executable code, saving significant time.

AI Output

```
import numpy as np
  from scipy.optimize import root_scalar
  # Define the function
  def f(x):
       return x**5 - x - 1
  # root_scalar will find a root within the given bracket [1, 2]
8
  # It uses a robust and efficient algorithm (Brent's method by
       default)
  sol = root_scalar(f, bracket=[1, 2])
  # Print the results
12
  print(f"Function: f(x) = x^5 - x - 1")
13
  print(f"Root found: {sol.root}")
  print(f"Function value at root: {f(sol.root)}")
  print(f"Iterations: {sol.iterations}")
```

In seconds, we have a working, efficient script. The next step, however, is the most important in any AI-assisted workflow: verification.

2.3 Step 3: Verification and Visualization

NEVER TRUST AN AI'S OUTPUT without verification. While the code *looks* correct, we must confirm the result and build our own intuition about the problem. A great way to do this is to have the AI generate more code to both verify the numerical result and visualize the function's behavior.

Prompt to AI

That's a good start. Now, modify the script to do two more things: 1. Add a formal verification step that prints whether the absolute value of the function at the found root is less than a tolerance of 1×10^{-12} .

2. Using Matplotlib, create a publication-quality plot of the function from x = -2 to x = 2. Clearly label the axes, add a title, and place a red dot on the plot at the location of the root.

The AI can now generate the code to produce the plot shown in Figure 2.1. When run, the script also outputs the crucial verification text:

```
Root found: 1.1673039782614187
Verifying root...
Value of f(root) is: 7.771561172376096e-16
Root verification successful: |f(root)| < 1e-12</pre>
```

Verification Output:

Root verification successful: |f(root)| < 1e-12 This is our proof. We have independently confirmed the AI's numerical result.



Figure 2.1: A plot of $f(x) = x^5 - x - 1$, generated via AI assistance. The unique real root is highlighted. The visual confirmation is a key part of the verification process.

The numerical result is confirmed to be highly accurate, and the plot visually confirms our analysis from Step 1. We see a single crossing point, located between 1 and 2.

2.4 Step 4: Dissemination

THE FINAL STEP IS TO COMMUNICATE OUR FINDINGS. This is where AI can be a powerful partner in generating formal mathematical text using LAT_EX.



The quintic polynomial $f(x) = x^5 - x - 1$ possesses a unique real root, which we determine numerically to be $x \ge 1.16730$.

When rendered, this gives us a perfect, publication-ready summary of our work:

Theorem 2.1 (Root of $x^5 - x - 1$). The quintic polynomial $f(x) = x^5 - x - 1$ possesses a unique real root, which we determine numerically to be $x \approx 1.16730$.

THIS SIMPLE EXAMPLE ENCAPSULATES THE ENTIRE WORKFLOW that this guide will explore in detail. We moved from a high-level **analysis**, to

Final step - communicate.

specific **code generation**, to rigorous **verification** and **visualization**, and finally to formal **dissemination**. At every stage, the human researcher directed the inquiry, and the AI provided the technical implementation. The following chapters will give you the skills to apply this powerful paradigm to your own research.

3____Background and History of AI in

Mathematics

We can only see a short distance ahead, but we can see plenty there that needs to be done.

— Alan Turing (1912 – 1954)

When you drink water, remember the source. 饮水思源。

— Chinese Proverb (中国谚语)

THE INTERSECTION OF ARTIFICIAL INTELLIGENCE AND MATHEMATICS has a rich and complex history that spans several decades, evolving from simple symbolic manipulation systems to sophisticated neural networks capable of engaging with abstract mathematical concepts. Understanding this evolution is essential for appreciating both the current capabilities and future potential of AI in mathematical research.

The earliest manifestations of AI in mathematics emerged in the 1960s with the development of computer algebra systems (CAS) such as MAC-SYMA, developed at MIT between 1968 and 1982. These systems represented the first serious attempt to automate mathematical computation beyond simple numerical calculations, focusing instead on symbolic manipulation of algebraic expressions. MACSYMA's ability to perform symbolic integration, differentiation, and algebraic manipulation laid the groundwork for what would become a fundamental tool in mathematical research.

The 1970s and 1980s saw significant advances in theorem proving systems, with notable developments including the Boyer-Moore theorem prover and the emergence of interactive proof assistants. These systems introduced the concept of computer-assisted proof verification, though they remained highly specialized tools requiring significant expertise to use effectively. The famous four-color theorem proof by Kenneth Appel and Wolfgang Haken in 1976, which relied heavily on computer verification, marked a watershed moment in the acceptance of computer-assisted mathematics, though it also sparked debates about the nature of mathematical proof that continue to this day.

The 1990s brought the development of more sophisticated computer algebra systems, including Mathematica (1988), Maple (1980s), and later MATLAB, which became increasingly accessible to working mathematicians. These systems not only automated routine calculations but also provided powerful visualization capabilities and programming environments that enabled new forms of mathematical exploration. The integration of these tools into mathematical education and research practices began to CAS Systems

theorem proving systems

reshape how mathematicians approached both teaching and discovery.

The turn of the millennium marked the beginning of a new era with the development of formal proof systems and proof assistants such as Coq, Lean, and Isabelle. These systems represented a significant advancement in the formalization of mathematical knowledge, allowing for the creation of machine-checkable proofs and the development of large libraries of verified mathematical results. The Flyspeck project, which formalized the proof of the Kepler conjecture, demonstrated the potential for large-scale formalization efforts in mathematics.

The 2010s witnessed the emergence of machine learning applications in mathematics, with systems beginning to recognize patterns in mathematical data and even generate conjectures. Notable developments included the use of deep learning for mathematical object recognition, the application of reinforcement learning to game theory and optimization problems, and the beginning of natural language processing applications in mathematical text analysis.

The current decade has been defined by the dramatic emergence of large language models, beginning with GPT-3 in 2020 and accelerating through subsequent developments including GPT-4, Claude, and specialized mathematical AI systems. These models represent a qualitative shift from previous AI applications in mathematics, offering capabilities that include natural language understanding of mathematical concepts, code generation, proof assistance, and even creative problem-solving approaches.

The development of these modern AI systems has been accompanied by significant advances in their mathematical capabilities. Systems like GPT-4 demonstrate remarkable fluency in mathematical communication, can solve complex problems across multiple domains, and can even engage in mathematical reasoning that approaches human-level performance in many areas. However, these capabilities come with important limitations and potential pitfalls that mathematicians must understand and navigate carefully.

Parallel to these developments in general AI, specialized mathematical AI systems have emerged that focus specifically on mathematical tasks. Systems like Lean's mathlib project, the Metamath database, and various automated theorem provers represent focused efforts to apply AI specifically to mathematical problems. These systems often achieve superior performance in their specialized domains compared to general-purpose AI, though they may lack the broad applicability and natural language interface of large language models.

The integration of AI into mathematical research has also been shaped by broader technological trends, including the availability of massive computational resources, the development of sophisticated programming languages and environments, and the growth of online mathematical communities and databases. These factors have created an ecosystem that supports both the development of AI tools and their adoption by working machine learning

formal proof systems

large language models

integration of AI into mathematical research

mathematicians.

Understanding this historical context is crucial for several reasons. It helps researchers appreciate the evolutionary nature of AI development and avoid both excessive optimism about current capabilities and unwarranted pessimism about future potential. It provides insight into the types of problems that AI systems are likely to handle well and those that may remain challenging. It also illuminates the ongoing debates about the role of computation in mathematical proof and discovery that continue to shape the field.

The trajectory of AI development in mathematics suggests that we are likely to see continued integration of these tools into standard mathematical practice, with potential developments including more sophisticated proof assistants, enhanced natural language interfaces for mathematical systems, and possibly even AI systems capable of independent mathematical discovery. However, this evolution will likely be accompanied by ongoing challenges related to verification, interpretation, and the maintenance of mathematical rigor.

The dream of automating mathematical reasoning is not new. The early history of AI, often called "Symbolic AI" or "Good Old-Fashioned AI" (GO-FAI), was built on the foundation of mathematical logic. The goal was to build systems that could reason from first principles, manipulating symbols according to formal rules. This philosophy gave rise to expert systems and, crucially for us, the first Computer Algebra Systems (CAS). Tools like Macsyma, developed at MIT in the 1960s, were direct descendants of this tradition. They are deterministic, logically consistent, and provably correct within their axiomatic systems. They are powerful calculators for symbolic mathematics.

For decades, this was the primary mode of AI in mathematics. The recent explosion in AI capabilities, however, comes from a different lineage: "Connectionist AI," which finds its modern expression in machine learning and deep neural networks. Unlike symbolic systems, these models are not programmed with explicit rules. Instead, they learn statistical patterns from vast amounts of data. The development of the **Transformer architecture** ¹ was a watershed moment, enabling the creation of Large Language Models (LLMs) like OpenAI's GPT series ², Anthropic's Claude, and Google's Gemini.

These LLMs are not logic engines. They are astonishingly sophisticated pattern-matchers and predictors. When you ask an LLM a question, it is not "thinking" or "reasoning" in the human sense; it is calculating the most probable sequence of words to form a plausible-sounding answer based on the patterns it learned from its training data (a significant portion of the internet). This distinction is the single most important concept for a mathematician to grasp. Continued integration of AI tools into standard mathematics

Early history of AI

¹ Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008, 2017

² OpenAI. ChatGPT, 2024. Model: GPT-4. Accessed on: 2024-05-21

Current Tools and Technologies

The real question is not whether machines think but whether men do. — **B.F. Skinner** (1904 – 1990)

Mathematics should serve practical applications, but practical applications should also inspire mathematics. 数学应当为实际应用服务,但实际应用也应当启发数学。

— Wu Wenjun (吴文俊)

THE CONTEMPORARY LANDSCAPE OF ARTIFICIAL INTELLIGENCE TOOLS available to mathematicians is diverse and rapidly evolving, encompassing everything from general-purpose large language models to specialized mathematical computation systems. Understanding the capabilities, limitations, and appropriate applications of these various tools is essential for effective integration into mathematical research workflows.

4.1 LARGE LANGUAGE MODELS AND CHATBOTS

LARGE LANGUAGE MODELS REPRESENT perhaps the most significant recent development in AI tools for mathematics. These systems, trained on vast corpora of text including mathematical literature, demonstrate remarkable capabilities in understanding and generating mathematical content, though they also exhibit important limitations that users must understand and accommodate.

GPT-4, developed by OpenAI, stands as one of the most capable generalpurpose AI systems currently available for mathematical applications. Its training encompassed a substantial portion of mathematical literature, enabling it to engage with concepts across virtually all areas of mathematics. The system demonstrates particular strength in problem-solving, proof sketching, and mathematical communication. When presented with a mathematical problem, GPT-4 can often provide multiple solution approaches, explain complex concepts in accessible language, and even identify potential errors in mathematical reasoning. However, users must remain aware that the system can occasionally produce plausible-sounding but incorrect mathematical statements, particularly in advanced or specialized areas.

Claude, developed by Anthropic, offers similar capabilities with some notable differences in approach and behavior. Claude tends to be more cautious about making definitive mathematical claims and is generally more explicit about uncertainty when dealing with complex or ambiguous problems. The system demonstrates strong capabilities in mathematical writing and communication, often producing well-structured explanations GPT-4

Claude

and proofs. Like GPT-4, Claude can assist with problem-solving across multiple mathematical domains, though it may be more conservative in its problem-solving approaches.

Google's Bard and other emerging LLMs provide additional options in the ecosystem, each with their own strengths and characteristics. The choice between these systems often depends on specific use cases, with some performing better for certain types of mathematical problems or communication tasks.

Specialized mathematical AI systems have emerged that focus specifically on mathematical applications. These systems often demonstrate superior performance in their targeted domains compared to generalpurpose LLMs, though they may lack the broad applicability and natural language interface that makes general systems accessible to many users.

These LLMs are your brainstorming partners, your coding assistants, and your LATEX scribes.

- Key Players: OpenAI's ChatGPT, Anthropic's Claude, Google's Gemini.
- **Best For:** Brainstorming, explaining concepts, generating code, debugging, and writing LATEX.
- The Catch: They can "hallucinate" incorrect information. Never trust a mathematical result from an LLM without independent verification.

4.2 Computer Algebra Systems

COMPUTER ALGEBRA SYSTEMS CONTINUE TO PLAY A CRUCIAL ROLE in mathematical research, and their integration with AI tools represents an important frontier in mathematical computation. Understanding the capabilities and appropriate applications of major CAS platforms is essential for effective mathematical research.

Mathematica, developed by Wolfram Research, provides one of the most comprehensive environments for symbolic computation. The system excels at symbolic manipulation, equation solving, and mathematical visualization. Its integration with Wolfram Alpha provides access to vast databases of mathematical knowledge and computational capabilities. For graphics generation, Mathematica offers sophisticated plotting capabilities that can produce publication-quality figures with precise control over mathematical notation and formatting. The system's notebook interface facilitates the integration of computation, visualization, and documentation in a single environment.

Exporting graphics from Mathematica requires attention to several factors to ensure scientific quality. The system supports multiple output formats including EPS, PDF, and high-resolution raster formats. For publication purposes, vector formats are generally preferred due to their scalability and clean rendering of mathematical notation. Proper axis laMathematica

Exporting graphics

Bard

beling, legend placement, and font consistency are crucial for producing scientifically acceptable graphics.

Maple, developed by Maplesoft, provides similar symbolic computation capabilities with its own strengths and interface characteristics. The system is particularly strong in certain areas of symbolic computation and offers excellent support for mathematical typesetting and document preparation. Maple's programming language provides powerful capabilities for developing custom mathematical applications and algorithms.

MATLAB, while primarily focused on numerical computation, provides important capabilities for mathematical research, particularly in applied mathematics and engineering applications. The system's strength lies in its extensive libraries for numerical analysis, signal processing, and scientific computing. MATLAB's plotting capabilities are particularly well-suited for data visualization and engineering applications, though they may require additional formatting for publication quality.

Python, while not traditionally considered a computer algebra system, has emerged as a powerful platform for mathematical computation through libraries such as SymPy, NumPy, and SciPy. The integration of Python with AI tools is particularly seamless, as many AI systems can generate Python code directly and execute it in real-time. This capability makes Python an excellent choice for mathematical research that involves both symbolic computation and AI integration.

CAS perform exact, error-free symbolic computation. Your choice will depend on your university's licenses, your field, and your personal preference.

• Wolfram Mathematica ¹: A cohesive, proprietary system with a powerful, consistent language and elegant notebook interface. Excellent for both symbolic and numerical work.

```
Al Output

    (* Define a function and compute its indefinite integral *)
    f[x_] := Exp[-x^2] * Sin[Pi*x]
    integral = Integrate[f[x], x]

    (* Generate a high-quality plot *)
    plotObject = Plot[f[x], {x, -3, 3},
        PlotLabel -> "Damped Sine Wave",
        AxesLabel -> {"x", "f(x)"},
        PlotStyle -> {Thick, Red},
        GridLines -> Automatic]

    (* Output the integral result *)
    Print["The integral is: ", integral]
```

 Maple ²: A long-standing commercial competitor to Mathematica, particularly lauded for its world-class symbolic computation engine, espe-

```
ing

Maple

ation

ystem

fers

a-

for

MATLAB

vides

ed

in its

tific

ed for

uire

Python

on

of

SymPy, NumPy, and SciPy

n

lity

lves
```

¹ Wolfram Research, Inc. Mathematica, Version 14.0, 2024

² Maplesoft. Maple. Waterloo, Ontario: Maplesoft, a division of Waterloo Maple Inc., 2024 cially in solving differential equations.

```
Al Output

# Solve differential equation symbolically
dolve(diff(y(x),x,x) + 2*diff(y(x),x) + y(x) = 0, y(x));

# Create publication-quality plot
plot(sin(x)*exp(-x), x = 0..2*Pi,
    title = "Damped Sine Wave",
    thickness = 2);
```

• MATLAB ³: Shines in numerical computing, matrix manipulations, and engineering applications. Its syntax is optimized for matrix and vector operations.

³ The MathWorks, Inc. MATLAB version R2024a, 2024

Al Output % Visualization of a function x = linspace(0, 2*pi, 1000); y = sin(x).*exp(-x); plot(x, y, 'LineWidth', 2); title('Damped Sine Wave'); xlabel('x'); ylabel('f(x)'); grid on;

• **SageMath** ⁴: A free, open-source project aiming to unify many math software packages under a Python interface.

Al Output



• **SymPy:** A pure Python library for symbolic math, perfect for projects that need to tightly integrate with other Python data science tools.

⁴ The Sage Developers. SageMath, the Sage Mathematics Software System (Version 10.3), 2024

Al Output

```
import sympy
  # Define a symbolic matrix
  t = sympy.Symbol('t')
  M = sympy.Matrix([
       [sympy.cos(t), -sympy.sin(t)],
       [sympy.sin(t), sympy.cos(t)]
7
  1)
8
  print("Matrix M(t):")
  sympy.pprint(M)
  # Compute the determinant
13
  det_M = M.det()
  print(f"\nDeterminant: {det_M}")
  print(f"Simplified Determinant: {sympy.simplify(det_M)}")
  # Compute the eigenvalues
  e_vals = M.eigenvals()
  print("\nEigenvalues:")
  sympy.pprint(e_vals)
```

• **Microsoft Excel:** Excellent for quickly organizing small datasets and performing simple calculations, but not suitable for formal, reproducible research.

4.3 AI IMAGE AND VIDEO GENERATION

THE EMERGENCE OF AI-POWERED IMAGE AND VIDEO GENERATION TOOLS has opened new possibilities for mathematical visualization and communication. These tools can create diagrams, illustrations, and even animations that support mathematical exposition and research.

Systems like DALL-E, Midjourney, and Stable Diffusion can generate mathematical diagrams and illustrations based on textual descriptions. While these systems are not specifically designed for mathematical applications, they can be effective for creating conceptual illustrations, geometric figures, and visual metaphors that support mathematical communication. However, users must be cautious about the accuracy of mathematically specific content generated by these systems.

Specialized mathematical visualization tools that incorporate AI capabilities are beginning to emerge, offering more precise control over mathematical content while maintaining the creative potential of AI-generated imagery. These tools show promise for creating sophisticated mathematical animations and interactive visualizations that can enhance both research and education. Tools like Midjourney and DALL-E 3 create images from text prompts. For mathematics, their primary use is for creating conceptual or artistic illustrations (e.g., "an artistic rendering of a fractal landscape"). They are generally useless for creating precise, data-driven plots or mathematical diagrams. For that, it is far better to use AI to generate the *code* for a plotting library like Matplotlib or TikZ.

4.4 Proof Assistants and Formal Systems

PROOF ASSISTANTS REPRESENT A SPECIALIZED but increasingly important category of AI tools for mathematics. These systems provide formal environments for constructing and verifying mathematical proofs, offering unprecedented levels of rigor and verification.

Lean, developed by Microsoft Research, has gained significant attention in the mathematical community for its modern design and growing library of formalized mathematics. The system provides a powerful type theory foundation for mathematical reasoning and has been used to formalize significant mathematical results. The Lean community has developed extensive libraries of mathematical knowledge, making it increasingly accessible for research applications.

Rocq Prover was formerly known as the Coq Proof Assistant Coq provides a mature environment for formal proof development with a long history of successful applications in both mathematics and computer science. The system's tactics-based approach to proof construction offers flexibility and power, though it requires significant investment in learning the system's methodology.

Isabelle, higher-order logic (HOL) theorem prover, offers another mature platform for formal mathematical reasoning, with particular strengths in certain areas of mathematics and logic. The system has been used for large-scale formalization projects and provides excellent support for mathematical automation.

The integration of these formal systems with AI tools represents an exciting frontier, with potential applications including AI-assisted proof construction, automated lemma discovery, and intelligent proof search. These developments suggest that formal methods may become increasingly accessible to working mathematicians.

A modern mathematician's toolkit is no longer just pen and paper. It's a suite of software designed for different tasks, from symbolic manipulation to text generation.

This is a rapidly growing area where AI and formal logic intersect. Proof assistants, also known as interactive theorem provers, are tools used to write formal proofs that can be automatically checked for correctness by a computer. They bridge the gap between human-readable proofs and machine-verifiable logic.

• Key Players: Lean ⁵, Coq, Isabelle/HOL.

Rocq Prover and Coq, https: //rocq-prover.org/.

Isabelle is a higher-order logic (HOL) theorem prover.

The future toolkit of the augmented mathematician.

⁵ Leonardo de Moura et al. The Lean theorem prover (system description). In 25th International Conference on Automated Deduction (CADE-25), volume 9195 of Lecture Notes in Computer Science, pages 378–388. Springer, 2015

- Use Case: They are used to formalize complex theorems, ensuring every step is logically sound. This has led to the verification of major results like the Four Color Theorem and the Kepler Conjecture.
- Role for Undergraduates: While an advanced topic, interacting with a proof assistant can provide profound insight into the rigorous foundations of mathematics. LLMs can sometimes help by generating boilerplate code for these systems, but the logical reasoning must be humandriven.

5 A Mathematician's Guide to How LLMs Work

All things are numbers.

— **Pythagoras** (c. 570 – c. 495 BC)

WHILE YOU DON'T NEED TO BUILD A CAR ENGINE TO DRIVE, a good driver knows what a transmission does and why oil is important. Similarly, to effectively use a Large Language Model (LLM) in mathematical research, you don't need to build one from scratch, but you do need a conceptual understanding of its mathematical engine.

The remarkable abilities of modern AI rest on a few profound mathematical ideas. At their heart, these systems translate the complexities of human language into the elegant, structured world of linear algebra and probability. This chapter provides a high-level overview of the core mathematical concepts that power LLMs, framed for a mathematical audience. For readers interested in a complete, implementation-level deep dive, we refer you to Appendix B.

5.1 Words as Vectors

THE FOUNDATIONAL INSIGHT OF MODERN AI is that words can be represented not as symbols, but as vectors in a high-dimensional vector space, typically \mathbb{R}^d where *d* might be several thousand. This is called an **embedding**. Every word in the AI's vocabulary is mapped to a unique vector.

This is not just a clever storage scheme; it's a geometric representation of meaning.

- Similarity is Distance: Words with similar meanings end up as vectors that are close to each other in this space. The "distance" between two words can be measured by the angle between their vectors, typically using the cosine similarity.
- **Relationships are Vectors:** The relationships between words can be captured by vector arithmetic. The most famous example is the vector equation:

$$\mathbf{v}_{\mathrm{king}} - \mathbf{v}_{\mathrm{man}} + \mathbf{v}_{\mathrm{woman}} \approx \mathbf{v}_{\mathrm{queer}}$$

This demonstrates that the geometric structure of the embedding space encodes deep semantic and analogical relationships.



Figure 5.1: A 2D projection of a word embedding space. Words with similar meanings, like 'cat' and 'feline', are close together. The vector from 'king' to 'queen' is similar to the vector from 'man' to 'woman', capturing the analogical relationship. Figure 5.2: A 2D projection of a word embedding space. Words with similar meanings, like 'cat' and 'feline', are close together. The vector from 'king' to 'queen' is similar to the vector from 'man' to 'woman', capturing the analogical relationship.

 $sim(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$

5.2 ATTENTION: A DYNAMIC, WEIGHTED AVERAGE

How DOES AN AI KNOW WHICH WORDS ARE IMPORTANT in a sentence? The answer is the **attention mechanism**, one of the key innovations of the Transformer architecture.

Mathematically, you can think of attention as a sophisticated, dynamic way of computing a weighted average. For each word in a sentence, the model calculates a set of "attention scores" that measure how relevant every other word is to it.

Given a sequence of input word vectors $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)$, the new representation for word *i*, let's call it \mathbf{y}_i , is computed as:

$$\mathbf{y}_i = \sum_{j=1}^n \alpha_{ij} \mathbf{x}_j$$

This is simply a weighted average of all the input vectors. The magic is in how the weights, α_{ij} , are calculated. They are determined on-the-fly based on the similarity between word *i* and every other word *j*.

The Core of Attention

The weights α_{ij} are calculated using matrix operations. The model learns three matrices: W_Q (for Queries), W_K (for Keys), and W_V (for Values).

- 1. Each input vector \mathbf{x}_i is transformed into a query $\mathbf{q}_i = \mathbf{x}_i \mathbf{W}_Q$ and a key $\mathbf{k}_i = \mathbf{x}_i \mathbf{W}_K$.
- 2. The "relevance" between word *i* and word *j* is the dot product of their query and key vectors: $s_{ij} = \mathbf{q}_i \cdot \mathbf{k}_j$.
- These scores are passed through a softmax function to ensure they sum to 1, creating the final weights: α_{ij} softmax(s_{i1}, s_{i2},..., s_{in})_j.

This entire process is just a series of matrix multiplications, making it incredibly fast and efficient to run on modern hardware like GPUs. The Transformer architecture, introduced in the paper "Attention Is All You Need," is the foundation for most modern LLMs.

The Core of Attention

_

5.3 PREDICTION AS A PROBABILITY PROBLEM

AT ITS CORE, AN LLM IS A MACHINE FOR PREDICTING THE NEXT WORD. When you give it a prompt, it calculates the most probable word to come next. Then it appends that word to the sequence and repeats the process, generating text one word at a time.

This is fundamentally a problem in probability theory. The model learns a massive conditional probability distribution, P(next word | previous words).

- **Training Objective:** The AI is trained on a vast corpus of text (like a large portion of the internet and digitized books). Its goal is to minimize a **loss function**. The most common is the **cross-entropy loss**, which measures the difference between the probability distribution the model predicts for the next word and the actual word that appeared in the training text.
- **Optimization:** The process of "learning" is simply a massive optimization problem. The model's parameters (the entries in its weight matrices, like W_Q, W_K, W_V) are adjusted iteratively using gradient descentbased methods (like the Adam optimizer) to minimize this loss function over the entire training dataset.

A neural network, then, can be thought of as a very complex, highdimensional function, f_{θ} (input text), where the parameters θ are the millions or billions of weights in the model. The training process is all about finding the optimal θ that makes the function a good predictor of human language.

THIS CONCEPTUAL FRAMEWORK—words as vectors, attention as a weighted average, and prediction as probability—is the mathematical foundation upon which modern AI is built. With this intuition in hand, you are now equipped to understand not just what AI can do for your mathematical research, but to develop a better sense of *how* it does it, and more importantly, where its limitations lie.

Effective Prompting Strategies

The art of asking the right question is more important than solving the problem. — **John Tukey** (1915-2000)

Learning without thinking is useless; thinking without learning is dangerous. 学而不思则罔,思而不学则殆。

— Confucius (孔子)

THE ART AND SCIENCE OF EFFECTIVE PROMPTING represents one of the most crucial skills for successfully integrating AI tools into mathematical research. Unlike traditional computational tools that require specific syntax and commands, AI systems respond to natural language instructions that can be crafted and refined to achieve desired outcomes. Understanding how to construct effective prompts is essential for maximizing the value of AI assistance while maintaining mathematical rigor and accuracy.

Effective prompting for mathematical AI systems requires understanding both the capabilities and limitations of the underlying models. These systems excel at pattern recognition, analogical reasoning, and the application of learned mathematical procedures, but they may struggle with novel constructions, highly specialized notation, or problems that require genuine creativity. Successful prompting strategies leverage the strengths of AI systems while compensating for their limitations through careful instruction design and iterative refinement.

The fundamental principle of effective mathematical prompting is clarity and specificity. Unlike human collaborators who can interpret ambiguous instructions based on context and shared understanding, AI systems benefit from explicit, detailed instructions that leave little room for misinterpretation. This requires prompt designers to be precise about their expectations, provide sufficient context for the problem domain, and clearly specify the desired format and level of detail for responses.

Context establishment forms the foundation of effective mathematical prompting. When presenting a problem to an AI system, it is crucial to provide sufficient background information to enable the system to understand not only what is being asked but also the mathematical framework within which the problem exists. This might include defining notation, specifying the mathematical domain, indicating the level of rigor expected, and providing relevant background theorems or definitions.

Consider the difference between a poorly constructed prompt:

```
Prompt: Solve this differential equation: y'' + 2y' + y = 0
```

and a well-constructed alternative:

Effective prompting for mathematical AI systems

clarity and specificity

Poorly constructed vs well-constructed prompts.



Prompt: I'm working on a second-order linear ordinary differential equation with constant coefficients: y'' + 2y' + y = 0. Please provide a complete solution including: (1) identification of the characteristic equation, (2) determination of the roots, (3) construction of the general solution, and (4) verification that the solution satisfies the original equation. Please use standard mathematical notation and show all intermediate steps.

The enhanced prompt provides several advantages: it establishes the mathematical context, specifies the expected solution format, requests verification steps, and indicates the desired level of detail. This approach significantly increases the likelihood of receiving a comprehensive and accurate response.

Getting high-quality, academic-style writing from an LLM is a skill. The default conversational style is often unsuitable for a formal paper. The key is to provide constraints on persona, format, and style.

Let's compare two prompts. The first is vague and will likely produce a conversational, high-level summary with bullet points, suitable for a blog post.

Prompt to AI

Before (Vague Prompt): Explain the Riemann Hypothesis.

The second prompt dictates the audience, tone, structure, length, and content, forcing the LLM to produce a much more sophisticated and useful draft.

Prompt to AI

After (Specific, Constrained Prompt): Act as a mathematics professor writing an introductory section for a survey paper aimed at advanced undergraduates.

Write a 300-word narrative introduction to the Riemann Hypothesis. The style should be formal and academic. Begin by situating the hypothesis in the context of the distribution of prime numbers, referencing the Prime Number Theorem as a starting point. Then, introduce the Riemann zeta function, $\zeta(s)$, and clearly state the hypothesis in terms of the location of its non-trivial zeros. Avoid using bullet points or subheadings. Ensure smooth transitions between concepts.

In general, models like Anthropic's Claude 3 Opus and OpenAI's GPT-4 are currently recognized for their ability to handle such nuanced writing tasks.

Iterative refinement represents another crucial strategy for effective prompting. Initial prompts rarely produce perfect results, and successful AI integration requires the ability to analyze responses, identify areas for improvement, and refine instructions accordingly. This process might involve asking follow-up questions, requesting clarification of specific Provide constraints on **persona**, format, and style.

Vague Prompt about Riemann Hypothesis

Specific, Constrained Prompt

Iterative refinement

steps, or providing additional constraints to guide the AI's reasoning.

The concept of role-playing in prompts has proven particularly effective for mathematical applications. By instructing the AI to adopt the persona of a specific type of mathematical expert—such as a topology specialist, number theorist, or applied mathematician—users can often obtain more focused and specialized responses. For example, beginning a prompt with

Prompt: You are an expert in algebraic topology. Please explain the concept of homology groups...

often yields more sophisticated and accurate explanations than generic requests.

Structured prompting techniques can significantly improve the quality of mathematical AI responses. This involves breaking complex problems into smaller, manageable components and requesting specific types of analysis for each component. For instance, when seeking help with a proof, a structured approach might request: (1) problem analysis and key insight identification, (2) proof strategy development, (3) detailed proof construction, and (4) verification and alternative approaches.

The use of examples in prompts can dramatically improve AI performance on mathematical tasks. By providing one or more examples of the desired response format, users can guide the AI toward producing outputs that match their expectations. This technique is particularly effective for standardized tasks such as problem-solving, proof formatting, or mathematical exposition.

Constraint specification is another powerful prompting strategy. By explicitly stating limitations or requirements—such as "use only elementary methods," "avoid advanced topology," or "provide a proof suitable for undergraduate students"—users can ensure that AI responses align with their specific needs and contexts.

Error handling and verification instructions should be integrated into mathematical prompts whenever possible. This might include requests for the AI to "double-check your work," "identify potential errors in your reasoning," or "provide alternative solution methods for verification." Such instructions help activate the AI's self-evaluation capabilities and can improve the reliability of responses.

The timing and sequencing of prompts can significantly impact their effectiveness. Rather than attempting to solve complex problems with a single, comprehensive prompt, users often achieve better results by breaking the problem-solving process into stages, using the output of one interaction to inform the next. This approach allows for course correction and refinement throughout the problem-solving process.

Advanced prompting techniques include chain-of-thought prompting, which explicitly requests that the AI show its reasoning process, and metaprompting, which involves asking the AI to analyze and improve its own responses. These techniques can be particularly valuable for complex mathematical problems that require multiple reasoning steps or creative role-playing

Structured prompting techniques

Constraint specification

Use of examples in prompts

Error handling and verification

problem-solving approaches.


In mathematics you don't understand things. You just get used to them. — John von Neumann (1903 – 1957)

Innovation is the soul of scientific and technological progress. 创新是科技进步的灵魂。

— Qian Xuesen (钱学森)

LARGE LANGUAGE MODEL CHAINING represents a sophisticated approach to AI integration that leverages the strengths of multiple AI systems or multiple interactions with the same system to achieve results that surpass what any single interaction might produce. This technique has particular relevance for mathematical research, where complex problems often require multiple types of analysis, verification steps, and refinement processes that can benefit from sequential AI assistance.

The fundamental concept of LLM chaining involves structuring AI interactions so that the output of one interaction becomes the input for subsequent interactions, creating a pipeline of AI-assisted analysis and refinement. This approach recognizes that while individual AI interactions may have limitations, carefully orchestrated sequences of interactions can overcome many of these limitations and produce more sophisticated, accurate, and comprehensive results.

In mathematical applications, LLM chaining can be particularly effective for complex problem-solving scenarios that require multiple types of expertise or analysis. For example, a challenging proof might benefit from an initial interaction focused on problem analysis and strategy development, followed by a second interaction that constructs the detailed proof, and a third interaction that verifies the logic and checks for errors. Each stage builds upon the previous work while bringing specialized focus to different aspects of the problem.

The design of effective LLM chains requires careful consideration of the logical flow between interactions and the specific contributions that each stage should make to the overall process. Successful chains often begin with broad analysis and strategy development, progress through detailed implementation or construction phases, and conclude with verification and refinement steps. This structure mirrors many natural problem-solving processes and allows each AI interaction to focus on tasks that are well-suited to its strengths.

One powerful application of LLM chaining in mathematics involves the iterative refinement of mathematical exposition. An initial interaction might focus on generating a rough draft of a mathematical explanation or proof, a second interaction might focus on improving clarity and organization, and a third might focus on ensuring mathematical rigor and accuracy. Create a pipeline of AI-assisted analysis and refinement.



This approach can produce explanations that are both mathematically sound and clearly communicated.

The concept of specialized chains involves using different AI systems or different prompting strategies at various stages of the chain to leverage particular strengths. For instance, one might use a system known for creative problem-solving in the initial stages, switch to a system known for rigorous analysis in the middle stages, and conclude with a system known for clear communication in the final stages. While this approach requires access to multiple AI systems, it can produce superior results for complex tasks.

Error detection and correction represent another important application of LLM chaining. A chain might include dedicated verification steps where the AI is specifically instructed to identify potential errors, inconsistencies, or gaps in reasoning from previous stages. This approach can significantly improve the reliability of AI-assisted mathematical work by building verification directly into the process.

The integration of computational tools into LLM chains opens additional possibilities for mathematical applications. A chain might begin with AI-generated problem analysis, proceed to computational verification using computer algebra systems, and conclude with AI-assisted interpretation of results. This hybrid approach combines the natural language capabilities of LLMs with the computational precision of specialized mathematical tools.

Practical implementation of LLM chaining requires attention to several technical considerations. The format and structure of outputs from each stage must be designed to serve as effective inputs for subsequent stages. This might involve requesting specific formatting, ensuring that key information is clearly highlighted, or including explicit instructions for how the output should be used in the next stage.

Documentation and tracking become crucial in complex LLM chains. Maintaining clear records of each interaction, the reasoning behind design decisions, and the evolution of ideas throughout the chain is essential for understanding the process and verifying the final results. This documentation also supports reproducibility and enables refinement of the chain for future applications.

Quality control in LLM chains requires particular attention to error propagation. Errors introduced in early stages can compound through subsequent interactions, potentially leading to sophisticated but fundamentally flawed results. Effective chains incorporate multiple verification points and error-checking mechanisms to identify and correct problems before they propagate through the entire process.

The evaluation of LLM chain outputs requires different approaches than the evaluation of single AI interactions. The final output must be assessed not only for accuracy and quality but also for internal consistency across the various stages of the chain. This might involve checking that conclusions align with initial analysis, that detailed work supports general claims, and that verification steps actually validate the proposed solutions.

Example LLM Chain for Proof Development

Stage 1: Problem Analysis

Prompt: Analyze the following theorem statement and identify key challenges, potential proof strategies, and relevant background knowledge needed ...

Stage 2: Strategy Development

Prompt: Based on the analysis from Stage 1, develop a detailed proof strategy including major steps, key lemmas that might be needed, and potential difficulties ...

Stage 3: Proof Construction

Prompt: Using the strategy from Stage 2, construct a detailed proof. Focus on logical rigor and clear argumentation ...

Stage 4: Verification

Prompt: Review the proof from Stage 3 and identify any logical gaps, unclear steps, or potential errors ...

Stage 5: Refinement

Prompt: Refine the proof addressing the issues identified in Stage 4, improving clarity and ensuring completeness ...

LLM Chaining is the process of using a sequence of prompts to iteratively build and refine a complex piece of work. Instead of trying to get a perfect result from a single prompt, you break the task down into logical steps, much like the human process of outlining, drafting, and revising. You can even use different LLMs for different steps (e.g., one for brainstorming, another for prose). This technique is especially powerful for refining a draft *without condensing it*.

Exercise 7.1 (Writing a Section on the Prime Number Theorem). Let's build a section of a paper step-by-step.

First, we ask the AI to act as a structured thinker and create a logical outline.

Prompt to AI

Prompt 1: Generate a logical outline for a section in an undergraduate paper about the historical development of the Prime Number Theorem. The outline should include the early empirical observations, the work of Legendre and Gauss, Chebyshev's bounds, Riemann's contribution, and the final proofs by Hadamard and de la Vallée Poussin.

Next, we take one part of the outline and ask the AI to flesh it out into a

Example LLM Chain for Proof Development prose draft.

Prompt to AI

Prompt 2: Using the first two points of the previous outline, write a draft of about 150 words covering the early empirical observations and the conjectures of Legendre and Gauss regarding the density of primes. Maintain a formal, academic tone.

This is the key chaining step. We take the draft and ask for refinement, not summarization.

Prompt to AI

Prompt 3: Act as a university writing tutor. Your task is to refine and enrich the following text. Do not shorten it. Your goals are to: 1. Improve the academic vocabulary and sentence structure. 2. Strengthen the transition between the empirical observations and the specific conjectures. 3. Ensure the tone is appropriate for a formal mathematical survey paper. Here is the text to refine: [Paste the text from the Step 2 output

here]

By following this process, you guide the AI's output at each stage, ensuring the final product aligns with your vision and quality standards.

8_AI-Assisted Mathematical Problem

Solving

The art of doing mathematics consists in finding that special case which contains all the germs of generality.

— David Hilbert (1862-1943)

Self-study is the method; diligence is the attitude; and innovation is the goal. 自学是方法,勤奋是态度,创新是目标。

— Hua Luogeng (华罗庚) (1910–1985)

THE INTEGRATION OF ARTIFICIAL INTELLIGENCE INTO MATHEMATI-CAL RESEARCH represents a paradigm shift in how we approach complex problems. From elementary optimization tasks to sophisticated partial differential equations, AI tools can serve as powerful computational partners that enhance our analytical capabilities. This chapter explores practical applications of AI in mathematical problem-solving, demonstrating how large language models can assist in formulating problems, generating code, and interpreting results across various mathematical domains.



Figure 8.1: Comprehensive workflow for AI-assisted mathematical problem solving. The process involves iterative cycles of human guidance, AI assistance, and verification, with multiple feedback loops ensuring mathematical rigor.

The systematic approach to AI-assisted mathematical problem solving follows the workflow illustrated in Figure 8.1. This process emphasizes the iterative nature of mathematical research, where initial AI suggestions are refined through multiple cycles of evaluation and improvement. The workflow highlights three critical principles: human oversight at every stage, rigorous verification of AI outputs, and transparent documentation of AI assistance.

8.1 Solving Ordinary Differential Equations

A POWERFUL DEMONSTRATION OF AI CAPABILITY is in solving Ordinary Differential Equations (ODEs), a fundamental task in all areas of science. AI can not only generate code to find numerical solutions but also plot them against known analytical solutions for verification.

Example 8.1. Solving a Simple ODE Let's ask an AI to solve and visualize the solution to a simple first-order ODE: y'(t) = -2ty with the initial condition y(0) = 1. This is a separable equation with a known analytical solution, $y(t) = e^{-t^2}$, making it perfect for verification.

Prompt to AI

Write a Python script to solve the ODE y'(t) = -2ty with y(0) = 1on the interval $t \in [0,2]$. Then, create a plot that shows both the numerical solution from SciPy's solve_ivp and the exact analytical solution $y(t) = \exp(-t^2)$.

Al Output

```
import numpy as np
   from scipy.integrate import solve_ivp
  import matplotlib.pyplot as plt
5 # Define the ODE function dy/dt = f(t, y)
  def ode_func(t, y):
       return -2 * t * y
  # Initial condition and time span
  y0 = [1]
  t_{span} = [0, 2]
  t_eval = np.linspace(t_span[0], t_span[1], 100)
  # Solve the ODE
  sol = solve_ivp(ode_func, t_span, y0, t_eval=t_eval)
  # Analytical solution for comparison
  y_analytical = np.exp(-t_eval**2)
  # Plotting the results
plt.figure(figsize=(8, 6))
plt.plot(sol.t, sol.y[0], 'b-', label='Numerical Solution')
plt.plot(t_eval, y_analytical, 'r--', label=r'Analytical: $e^{-
       t^2}$')
   plt.title("Solution of y'(t) = -2ty with y(0)=1")
   plt.xlabel('t'); plt.ylabel('y(t)')
  plt.legend(); plt.grid(True)
   plt.show()
```

Simple first order ODE.

This simple workflow demonstrates the core process: the AI generates code for both the numerical solution and the analytical verification, providing immediate confirmation of the result.

Exercise 8.1. Analyzing the Lotka-Volterra System The Lotka-Volterra equations model predator-prey dynamics and produce fascinating periodic orbits in the phase plane. Analyzing this system requires finding equilibria, checking their stability via linearization, and plotting trajectories.

Prompt to AI

I want to analyze the Lotka-Volterra predator-prey system

$$\dot{x} = ax - bxy, \quad \dot{y} = -cy + dxy.$$

Help me find the equilibrium points, analyze their stability using the Jacobian, and create a comprehensive phase portrait showing the direction field and several trajectories with different initial conditions. Use a = 1, b = 1, c = 1.5, d = 0.75.

Task:

- 1. Use the prompt above with a capable AI.
- 2. The AI should provide a mathematical analysis of the equilibrium points (extinction and coexistence) and their stability.
- 3. It should then generate a Python script to produce a phase portrait.
- Run the code and verify it produces a plot showing the stable center and periodic orbits.
- 5. Can you identify the stable center in the plot? Does it match the AI's analytical calculation?

Expected Skills: Equilibrium analysis, Jacobian matrices, phase plane analysis, numerical integration of ODEs.

8.2 Solving Partial Differential Equations

PARTIAL DIFFERENTIAL EQUATIONS (PDEs) ARE NOTORIOUSLY DIF-FICULT to solve analytically. AI-assisted numerical methods provide a powerful toolkit for exploring their solutions. A revolutionary modern approach is the Physics-Informed Neural Network (PINN), which embeds the PDE directly into the loss function of a neural network.

Example 8.2. Solving the Heat Equation with PINNs The heat equation,

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

Lotka-Volterra equations

Heat equation example

, is a canonical parabolic PDE. A PINN can solve it without traditional mesh-based methods. The loss function combines three terms:

Loss function

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{physics}} + \mathcal{L}_{\text{boundary}} + \mathcal{L}_{\text{initial}}$$

Prompt to Al

I want to solve the 1D heat equation using a PINN. The equation is $u_t = \alpha u_{xx}$ with initial condition $u(x,0) = \sin(\pi x)$ and zero boundary conditions. Provide a simple implementation using PyTorch showing the key components: network definition, physics loss, and training loop.

AI Output

```
import torch
import torch.nn as nn
import numpy as np
class HeatPINN(nn.Module):
    def __init__(self):
        super(HeatPINN, self).__init__()
        self.net = nn.Sequential(
            nn.Linear(2, 20), nn.Tanh(),
            nn.Linear(20, 20), nn.Tanh(),
            nn.Linear(20, 1)
        )
    def forward(self, x, t):
        inputs = torch.cat([x, t], dim=1)
        return self.net(inputs)
    def physics_loss(self, x, t, alpha=0.01):
        x.requires_grad_(True)
        t.requires_grad_(True)
        u = self.forward(x, t)
        u_t = torch.autograd.grad(u, t, torch.ones_like(u),
             create_graph=True)[0]
        u_x = torch.autograd.grad(u, x, torch.ones_like(u),
            create_graph=True)[0]
        u_xx = torch.autograd.grad(u_x, x, torch.ones_like(u_x)
             , create_graph=True)[0]
        return u_t - alpha * u_xx
# Training loop
model = HeatPINN()
optimizer = torch.optim.Adam(model.parameters(), lr=0.001)
for epoch in range(1000):
    # Generate training points
```

```
x_train = torch.rand(100, 1)
t_train = torch.rand(100, 1)
t_train = torch.rand(100, 1)

# Physics loss
physics_residual = model.physics_loss(x_train, t_train)
loss = torch.mean(physics_residual**2)
# Add boundary and initial losses here...
# Add boundary and initial losses here...
# optimizer.zero_grad()
loss.backward()
optimizer.step()
# if epoch % 200 == 0:
# print(f'Epoch {epoch}, Loss: {loss.item():.6f}')
```

This example demonstrates the key innovation of PINNs: embedding physical laws directly into neural network training through automatic differentiation.

KdV Soliton example

Exercise 8.2. Simulating Soliton Collisions Nonlinear PDEs like the Korteweg-de Vries (KdV) equation, $u_t + 6uu_x + u_{xxx} = 0$, have remarkable soliton solutions that maintain their shape after collision.

Prompt to AI

I want to explore soliton solutions of the KdV equation. Implement a spectral method in Python to solve the equation for an initial condition consisting of two solitons of different amplitudes. Generate a space-time plot that visualizes their collision and shows that they emerge with their shapes and speeds intact.

Task:

- 1. Use the prompt to generate a Python script using scipy.fft.
- 2. The AI should set up an initial condition with two sech² profiles.
- 3. It should implement time-stepping where spatial derivatives are computed in Fourier space.
- 4. Run your code and verify it produces a collision visualization showing phase shift after interaction.

Expected Skills: Spectral methods, FFT, nonlinear wave equations, advanced visualization.

8.3 DATA ANALYSIS AND STATISTICS

MODERN DATA ANALYSIS BENEFITS ENORMOUSLY from AI assistance, particularly for exploratory data analysis, statistical modeling, and visualization. AI can help translate research questions into appropriate statistical tests and generate comprehensive analysis pipelines.

Example 8.3. Exploratory Data Analysis Let's demonstrate how AI can assist with a complete data analysis workflow for a dataset with demographic and satisfaction variables.

Prompt to AI

I have a dataset with columns 'age', 'income', 'education_years', and 'satisfaction_score'. Help me perform exploratory data analysis including summary statistics, correlation analysis, and visualizations. Also implement a simple regression model to predict satisfaction_score.

Al Output

```
import pandas as pd
  import numpy as np
  import matplotlib.pyplot as plt
4 import seaborn as sns
  from sklearn.linear_model import LinearRegression
  from sklearn.model_selection import train_test_split
  # Load and examine data
  print("Dataset Overview:")
  print(data.describe())
<sup>12</sup> # Correlation matrix
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
      center=0)
  plt.title('Correlation Matrix')
  plt.show()
  # Regression analysis
  X = data[['age', 'income', 'education_years']]
  y = data['satisfaction_score']
  X_train, X_test, y_train, y_test = train_test_split(X, y,
      test_size=0.2)
  model = LinearRegression()
  model.fit(X_train, y_train)
  print(f"R^2 Score: {model.score(X_test, y_test):.3f}")
  print("Coefficients:")
```

Data analysis example

```
for feature, coef in zip(X.columns, model.coef_):
    print(f" {feature}: {coef:.4f}")
```

This workflow demonstrates how AI can quickly generate a complete analysis pipeline from data loading through model interpretation.

times series example

Exercise 8.3. Time Series Analysis with ARIMA Time series analysis is crucial in many fields. Practice using AI to implement and tune an ARIMA model.

Prompt to AI

I have daily stock price data for the past 2 years. Help me implement an ARIMA model to forecast the next 30 days. Include: 1) stationarity testing with ADF test, 2) finding optimal parameters using AIC, 3) fitting the model, 4) generating forecasts with confidence intervals, and 5) visualizing results.

Expected Skills: ARIMA model components, stationarity testing, parameter selection, forecast evaluation.

8.4 Combinatorics and Graph Theory

AI EXCELS AT HELPING WITH COMBINATORIAL PROBLEMS, from basic counting to complex optimization problems on graphs and networks.

Example 8.4. Social Network Analysis Graph theory provides powerful tools for analyzing social networks, transportation systems, and many other networked structures.

Prompt to AI

Help me analyze a social network graph. Create a random graph with some community structure, compute basic network metrics (degree centrality, betweenness centrality, clustering coefficient), detect communities, and visualize the results with meaningful colors and layouts.

Al Output

```
import networkx as nx
import numpy as np
import matplotlib.pyplot as plt
import community as community_louvain

f
    # Create a graph with community structure
    G = nx.karate_club_graph() # Classic example
```

social network example

```
# Compute network metrics
   degree_centrality = nx.degree_centrality(G)
  betweenness_centrality = nx.betweenness_centrality(G)
  clustering = nx.clustering(G)
  # Community detection
  partition = community_louvain.best_partition(G)
  # Visualization
  fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 5))
  # Plot 1: Node size by degree centrality
  pos = nx.spring_layout(G, seed=42)
prode_sizes = [3000 * degree_centrality[node] for node in G.
       nodes()]
  nx.draw(G, pos, node_size=node_sizes, node_color='lightblue',
          with_labels=True, ax=ax1)
  ax1.set_title('Node Size = Degree Centrality')
  # Plot 2: Communities
  community_colors = [partition[node] for node in G.nodes()]
  nx.draw(G, pos, node_color=community_colors, cmap='Set3',
         node_size=300, with_labels=True, ax=ax2)
  ax2.set_title('Detected Communities')
   plt.show()
   print(f"Graph has {G.number_of_nodes()} nodes and {G.
       number_of_edges()} edges")
   print(f"Average clustering: {nx.average_clustering(G):.3f}")
   print(f"Number of communities: {len(set(partition.values()))}")
```

This example shows how AI can help analyze complex network structures with just a few lines of code.

Exercise 8.4. Traveling Salesman Problem The TSP is a classic optimization problem that demonstrates the difference between exact and heuristic algorithms.

Prompt to Al

Help me implement and compare different approaches to solve the Traveling Salesman Problem for 15 cities. Include: 1) nearest neighbor heuristic, 2) 2-opt improvement, and 3) genetic algorithm. Visualize the tours and compare solution quality and runtime.

Expected Skills: Heuristic algorithms, local search, metaheuristics, computational complexity analysis.

Traveling Salesman Problem

8.5 Pure Mathematics: Number Theory

AI CAN ASSIST EVEN WITH PURE MATHEMATICS, helping to explore patterns, test conjectures, and implement number-theoretic algorithms.

Example 8.5. Prime Number Exploration Let's explore patterns in prime numbers using computational methods.

Prompt to AI

Help me explore prime number patterns. Generate the first 1000 primes, analyze gaps between consecutive primes, find twin primes, and visualize the prime counting function $\pi(x)$ compared to the approximation $x/\ln(x)$.

Al Output

```
import numpy as np
   import matplotlib.pyplot as plt
   from collections import Counter
   def sieve_of_eratosthenes(limit):
       is_prime = [True] * (limit + 1)
       is_prime[0] = is_prime[1] = False
       for i in range(2, int(limit**0.5) + 1):
           if is_prime[i]:
               for j in range(i*i, limit + 1, i):
                   is_prime[j] = False
       return [i for i in range(2, limit + 1) if is_prime[i]]
   # Generate first 1000 primes
   primes = sieve_of_eratosthenes(10000)[:1000]
   # Analyze prime gaps
   gaps = [primes[i+1] - primes[i] for i in range(len(primes)-1)]
  # Find twin primes
22
  twin_primes = [(primes[i], primes[i+1]) for i in range(len(
       primes)-1)
                  if primes[i+1] - primes[i] == 2]
  # Prime counting function vs approximation
  x_values = range(10, 1000)
   pi_x = [sum(1 for p in primes if p <= x) for x in x_values]</pre>
  approximation = [x / np.log(x) \text{ for } x \text{ in } x_values]
  # Visualizations
  fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize=(12,
        8))
```

```
ax1.hist(gaps, bins=20, alpha=0.7)
   ax1.set_title('Distribution of Prime Gaps')
  ax2.scatter(range(len(twin_primes)), [p[0] for p in twin_primes
       ], s=10)
  ax2.set_title('Twin Primes')
  ax3.plot(x_values, pi_x, 'b-', label='pi(x)')
  ax3.plot(x_values, approximation, 'r--', label='x/ln(x)')
  ax3.set_title('Prime Counting Function')
42
ax3.legend()
  ax4.scatter(primes[:100], [1]*100, s=5)
45
  ax4.set_title('First 100 Primes')
  plt.tight_layout()
  plt.show()
  print(f"1000th prime: {primes[999]}")
  print(f"Twin primes found: {len(twin_primes)}")
   print(f"Most common gap: {Counter(gaps).most_common(1)[0]}")
```

This exploration reveals patterns in prime distribution and demonstrates computational number theory techniques.

Exercise 8.5. Investigating the Collatz Conjecture

The Collatz Conjecture is one of the most famous unsolved problems in mathematics.

Prompt to AI

Investigate the Collatz Conjecture (3n + 1 problem). Create functions to compute Collatz sequences, analyze stopping times for numbers 1-1000, find the number with the longest sequence, visualize the distribution of stopping times, and create a tree showing how different numbers converge.

Expected Skills: Recursive algorithms, statistical analysis of sequences, mathematical conjectures, data visualization.

8.6 Advanced Prompting Techniques

THE EFFECTIVENESS OF AI ASSISTANCE depends critically on the quality and specificity of prompts. Sophisticated mathematical problems require carefully crafted queries that build understanding through sequences of related questions. Collatz Conjecture

Layered Prompting Strategy

Rather than attempting to solve complex problems in a single query, effective prompting involves building understanding incrementally.

Example 8.6. Creative Mathematical Discovery Historical perspective can illuminate modern mathematical concepts. Consider this creative prompt:

Prompt to AI

Imagine you are Euler discovering the relationship between exponentials and trigonometric functions. Walk me through the thought process that might have led to $e^{ix} = \cos(x) + i\sin(x)$. Then show me how this connects to Fourier analysis and create a visualization showing the unit circle representation.

This approach demonstrates how framing exploration as historical discovery provides both mathematical rigor and intuitive understanding.

Exercise 8.6. Advanced Mathematical Connections Choose one of these advanced prompting challenges:

Option A: Hopf Bifurcation Analysis

Prompt to AI

I'm studying Hopf bifurcations in dynamical systems. Explain the mathematical conditions for a Hopf bifurcation, show me how to detect it using eigenvalue analysis, and demonstrate with the van der Pol oscillator including bifurcation diagrams.

Option B: Golden Ratio Exploration

Prompt to AI

Explore mathematical connections of the golden ratio ϕ . Show how it appears in Fibonacci sequences, continued fractions, regular pentagons, and optimization. Create visualizations demonstrating these connections.

Expected Skills: Advanced prompting strategies, interdisciplinary mathematical connections, mathematical communication.

8.7 Optimization and Machine Learning

MODERN OPTIMIZATION TECHNIQUES form the backbone of machine learning and can be applied across many mathematical domains.

Example 8.7. Portfolio Optimization Consider the classical problem where an investor seeks to maximize returns while minimizing risk.

Euler's formula

Hopf bifurcation

Golden ratio and Fibonacci sequence

portfolio optimization problem

Prompt to Al

I need to solve a portfolio optimization problem. I have 4 assets with expected returns [0.12, 0.08, 0.15, 0.10] and a covariance matrix. Find portfolio weights that minimize risk for a target return of 0.11. Formulate as quadratic programming and solve.

Al Output

```
import numpy as np
  import cvxpy as cp
  # Problem data
  mu = np.array([0.12, 0.08, 0.15, 0.10]) # Expected returns
  Sigma = np.array([[0.04, 0.01, 0.02, 0.01],
                     [0.01, 0.02, 0.00, 0.01],
                     [0.02, 0.00, 0.06, 0.02],
                     [0.01, 0.01, 0.02, 0.03]]) # Covariance
                        matrix
  r_target = 0.11
  # Optimization variables
  w = cp.Variable(4)
   # Objective: minimize portfolio variance
  objective = cp.Minimize(0.5 * cp.quad_form(w, Sigma))
  # Constraints
  constraints = [
      mu.T @ w == r_target, # Target return
      cp.sum(w) == 1, # Weights sum to 1
       w >= 0
                            # No short selling
  ]
23
  # Solve
25
   prob = cp.Problem(objective, constraints)
  prob.solve()
   print(f"Optimal weights: {w.value}")
   print(f"Portfolio return: {mu.T @ w.value:.4f}")
  print(f"Portfolio risk: {np.sqrt(2 * prob.value):.4f}")
```

This demonstrates how AI correctly identifies the problem type and provides both mathematical formulation and working implementation.

Exercise 8.7. Neural Network Training from Scratch Understanding optimization in machine learning provides insights into both fields.

Neural Network Training

Prompt to Al

Help me implement a neural network from scratch for binary classification. Include forward propagation, backpropagation, and compare SGD, momentum, and Adam optimizers. Use a synthetic dataset and visualize convergence behavior.

Expected Skills: Automatic differentiation, gradient-based optimization, machine learning fundamentals.

8.8 Capstone Research Projects

CHOOSE ONE RESEARCH-LEVEL PROJECT to explore throughout the semester, applying AI assistance techniques learned in this chapter.

Project Options

- 1. **Mathematical Biology:** Model epidemic spread, population dynamics, or tumor growth
- 2. **Financial Engineering:** Develop quantitative trading strategies using mathematical models
- 3. **Computational Physics:** Simulate physical systems using numerical methods
- 4. **Cryptography:** Explore number-theoretic algorithms in modern cryptography
- 5. Operations Research: Solve real-world optimization problems
- 6. **Data Science:** Apply advanced statistical methods to analyze complex datasets

Project Requirements

For your chosen project:

- Literature Review: Research existing mathematical approaches
- **Implementation:** Code solutions using AI assistance, documenting prompting strategies
- Analysis: Validate results, perform sensitivity analysis, compare approaches
- Visualization: Create compelling graphics communicating findings
- **Reflection:** Discuss AI's role in your mathematical discovery process

Prompting Best Practices

Throughout your projects, apply these strategies:

- Start Simple: Begin with basic versions before adding complexity
- **Be Specific:** Include mathematical notation, parameters, expected outputs
- Iterate: Build understanding through sequences of related questions
- Validate: Always request verification methods and analytical solutions
- Explain: Request explanations of concepts, not just code

8.9 CONCLUSION AND FUTURE DIRECTIONS

THE EXPLORATION OF AI-ASSISTED MATHEMATICAL PROBLEM-SOLVING reveals both tremendous potential and inherent limitations. Throughout this chapter, we have seen how large language models can serve as powerful mathematical partners, capable of translating conceptual ideas into rigorous formulations, implementing sophisticated algorithms, and providing insights that bridge different mathematical domains.

Several key themes emerge from our analysis. First, AI tools excel at connecting different mathematical areas, revealing unexpected relationships between seemingly disparate fields. The progression from simple optimization problems to complex interdisciplinary models illustrates how AI can help synthesize knowledge across differential equations, probability theory, numerical analysis, and applied mathematics.

Second, the combination of symbolic and numerical approaches becomes particularly powerful when mediated by AI assistance. Physics-Informed Neural Networks exemplify this synthesis, embedding mathematical constraints directly into machine learning frameworks while maintaining flexibility for complex problems.

Third, the role of visualization and computational exploration becomes central to mathematical understanding. AI-generated code consistently emphasizes graphical representation and numerical experimentation, supporting intuitive development that complements formal analysis.

Looking forward, advances in multimodal AI systems and integration with computer algebra systems will enhance mathematical AI capabilities. However, the fundamental importance of mathematical understanding, creativity, and critical thinking remains unchanged. AI tools amplify human mathematical capabilities but cannot replace the deep conceptual insights and careful reasoning that drive mathematical discovery.

The exercises in this chapter provide a foundation for this collaboration, demonstrating how AI can serve as an intelligent mathematical partner. As you work through these exercises, remember that the goal is not just to

generate working code, but to develop mathematical intuition, problemsolving skills, and the ability to leverage AI tools effectively in your future mathematical endeavors.

The most successful applications of AI in mathematics will continue to emphasize the collaborative relationship between human insight and computational power, where mathematicians provide direction, interpretation, and validation while AI systems contribute pattern recognition, implementation expertise, and the ability to rapidly explore mathematical landscapes.

Using AI for Problem Solving

A problem well stated is a problem half solved.

- Charles Kettering (1876 –1958)

In mathematics there are no shortcuts to anywhere. 数学中没有捷径可走。

— Hua Luogeng (华罗庚) (1910-1985)

THE APPLICATION OF ARTIFICIAL INTELLIGENCE to mathematical problemsolving represents one of the most immediate and practical benefits of AI integration in mathematics. Modern AI systems demonstrate remarkable capabilities in approaching mathematical problems across diverse domains, from elementary calculus to advanced topics in algebra, analysis, and discrete mathematics. However, effective utilization of these capabilities requires understanding not only what AI systems can do but also how to structure interactions to maximize their problem-solving potential while maintaining mathematical rigor.

AI systems excel at pattern recognition and the application of learned solution strategies to new problems. When presented with a mathematical problem, these systems can often identify relevant techniques, suggest solution approaches, and even carry out detailed solution procedures. This capability makes them valuable collaborators in the problem-solving process, offering fresh perspectives and alternative approaches that might not be immediately apparent to human mathematicians.

The strength of AI in mathematical problem-solving lies particularly in its ability to draw connections between seemingly disparate areas of mathematics. AI systems trained on broad mathematical corpora can recognize when techniques from one domain might be applicable to problems in another, suggesting interdisciplinary approaches that human experts might overlook. This cross-pollination of ideas can lead to elegant solutions and new insights into mathematical relationships.

However, AI problem-solving capabilities are not without limitations. These systems may struggle with problems requiring genuine creativity, novel constructions, or deep conceptual insights that go beyond learned patterns. They may also have difficulty with problems involving very recent mathematical developments or highly specialized notation that was not well-represented in their training data.

The most effective approach to AI-assisted problem-solving involves treating the AI as a sophisticated collaborator rather than an infallible oracle. This means using AI to generate ideas, suggest approaches, and carry out routine calculations while maintaining human oversight of the overall problem-solving strategy and verification of results. This collaborative approach leverages the complementary strengths of human insight and AI limitations

9

computational capability.

Problem decomposition represents one of the most powerful techniques for AI-assisted mathematical problem-solving. Complex problems can often be broken down into smaller, more manageable components that are well-suited to AI assistance. This might involve identifying subproblems, isolating key calculations, or separating conceptual issues from computational ones. Each component can then be addressed through focused AI interactions, with human oversight ensuring that the components are properly integrated into a complete solution.

The iterative nature of mathematical problem-solving aligns well with AI capabilities. Problems often require multiple attempts, refinements, and approaches before reaching a satisfactory solution. AI systems can assist at each stage of this process, suggesting modifications to unsuccessful approaches, identifying errors in reasoning, and proposing alternative strategies when initial attempts fail.

Verification and validation represent crucial aspects of AI-assisted problem-solving. While AI systems can generate solutions and approaches, human oversight is essential for ensuring that these solutions are correct, complete, and appropriately rigorous. This might involve checking calculations, verifying logical steps, or testing solutions against known results or special cases.

The integration of AI problem-solving with computational tools creates powerful hybrid approaches. AI can suggest analytical strategies while computational tools handle numerical verification, or AI can interpret computational results and suggest theoretical explanations. This combination of natural language reasoning and computational precision can be particularly effective for applied mathematics problems.

Documentation of AI-assisted problem-solving processes is important for several reasons. It supports reproducibility, enables learning from successful approaches, and provides transparency about the role of AI in the solution process. This documentation should include not only the final solution but also the reasoning process, alternative approaches considered, and verification steps taken.

Example: AI-Assisted Optimization Problem

Problem: Find the maximum value of $f(x, y) = x^2 + y^2$ subject to the constraint $x^2 + 2y^2 = 1$.

AI Interaction 1: "I have an optimization problem with constraint. Please suggest solution approaches and identify the most appropriate method."

AI Response: Suggests Lagrange multipliers, geometric interpretation, and substitution methods. Recommends Lagrange multipliers for systematic approach.

AI Interaction 2: "Using Lagrange multipliers, please set up the system of equations and solve for critical points."

iteration Verification and validation Computational tool integration

Problem decomposition

documentation

AI Response: Establishes $\nabla f = \lambda \nabla g$, solves the system, identifies critical points.

AI Interaction 3: "Please verify the solution and check boundary conditions."

AI Response: Confirms calculations, evaluates function at critical points, identifies maximum.

In the early stages of research, AI can help you explore a topic, formulate questions, and test ideas.

Example 9.1 (Brainstorming Conjectures). You can use an AI to explore a new mathematical area or generate ideas.

Prompt to Al

I'm starting a project on spectral graph theory. Can you suggest some interesting, accessible research questions or conjectures related to the Laplacian matrix of a graph for an undergraduate?

Al Output

Certainly. Here are a few accessible directions:

- 1. Relationship between Graph Structure and Eigenvalues: How do small changes to a graph (e.g., adding/removing an edge) affect the algebraic connectivity (λ_2)? Can you characterize the graphs for which adding an edge causes the largest possible increase in λ_2 ?
- 2. **Cospectral Graphs:** Investigate methods for constructing nonisomorphic graphs that share the same Laplacian spectrum (cospectral mates). This is a classic problem with many open questions.

Example 9.2 (Finding Counterexamples). Als can be surprisingly effective at finding counterexamples to nascent conjectures.

Conjecture 9.1 (Incorrect). For any two $n \times n$ matrices A and B, det(A + B) = det(A) + det(B).

Prompt to AI

Find a counterexample to the conjecture that for any two $n \times n$ matrices A and B, det(A + B) = det(A) + det(B).

Al Output

Let
$$A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$
 and $B = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$. Then $det(A) = 0$ and $det(B) =$

Example: AI-Assisted Optimization Problem

Brainstorming

Counterexamples

0, so det(A) + det(B) = 0. However, $A + B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I.$ det(A + B) = det(I) = 1. Since $1 \neq 0$, this is a valid counterexample.

10 Code Generation and Computation

The computer was born to solve problems that did not exist before.

— Bill Gates (1955 -)

Let the computer do what computers do best, and let humans do what humans do best.

让计算机做计算机最擅长的,让人做人最擅长的。

— Wu Wenjun (吴文俊) (1919 – 2017)

THE ABILITY OF MODERN AI SYSTEMS TO GENERATE CODE represents a transformative capability for mathematical research, offering new possibilities for implementing algorithms, creating visualizations, and integrating computational tools into research workflows. This capability extends beyond simple code generation to include the creation of complete computational environments, the integration of multiple programming languages and tools, and the development of custom mathematical applications.

AI systems demonstrate remarkable proficiency in generating code across multiple programming languages commonly used in mathematical research. Whether working in Python, MATLAB, Mathematica, or other mathematical computing environments, AI can translate mathematical concepts into executable code, implement algorithms from theoretical descriptions, and create custom tools for specific research applications. This capability significantly reduces the barrier to entry for computational mathematics and enables researchers to focus on mathematical concepts rather than programming syntax.

The integration of AI code generation with mathematical research workflows creates powerful synergies. Researchers can describe mathematical procedures in natural language and receive working implementations, request modifications to existing code through conversational interfaces, and even generate entire analysis pipelines from high-level descriptions. This natural language programming capability makes computational tools more accessible to mathematicians who may not have extensive programming backgrounds.

Python has emerged as a particularly important language for AIassisted mathematical computing due to its extensive ecosystem of mathematical libraries and the fluency of AI systems in Python code generation. Libraries such as NumPy, SciPy, SymPy, and Matplotlib provide comprehensive capabilities for numerical computation, symbolic mathematics, and visualization. AI systems can effectively utilize these libraries to create sophisticated mathematical applications and analysis tools.

The generation of publication-quality graphics represents a particularly valuable application of AI code generation. AI systems can create code that produces professional mathematical visualizations, including plots,

Python

publication-quality graphics

diagrams, and animations that meet publication standards. This capability includes proper handling of mathematical notation, appropriate scaling and formatting, and integration with LATEX document preparation systems.

Verification and testing of AI-generated code require special attention in mathematical applications. While AI systems can produce syntactically correct and functionally appropriate code, mathematical accuracy cannot be assumed without verification. Effective practices include testing code against known results, implementing multiple solution approaches for comparison, and including assertion statements and error checking in generated code.

The iterative refinement of AI-generated code mirrors the iterative nature of mathematical problem-solving. Initial code generation provides a foundation that can be refined through subsequent interactions, with modifications addressing performance issues, accuracy concerns, or feature enhancements. This iterative approach allows researchers to develop sophisticated computational tools through collaboration with AI systems.

Integration with existing computational environments represents another important aspect of AI code generation. AI systems can generate code that interfaces with established mathematical software, imports data from various sources, and exports results in appropriate formats. This integration capability enables researchers to incorporate AI-generated components into existing research workflows without requiring complete system overhauls.

The documentation and explanation of AI-generated code is crucial for research reproducibility and collaboration. AI systems can generate not only functional code but also accompanying documentation, comments, and explanations that help researchers understand and maintain the generated implementations. This documentation capability supports the sharing of computational tools and methods within the research community.

Performance optimization represents an area where AI code generation can be particularly valuable. AI systems can suggest optimizations for mathematical algorithms, identify inefficient code patterns, and recommend alternative implementations that better leverage available computational resources. This capability is especially important for research involving large-scale computations or real-time applications.

This is where AI can save you countless hours. Its ability to generate, debug, and translate code is one of its most reliable features.

The integration of AI-generated code with mathematical research workflows extends beyond simple function implementation to encompass entire research pipelines. AI systems can generate code that handles data preprocessing, implements multiple analysis methods, performs statistical tests, and generates comprehensive reports. This capability enables researchers to rapidly prototype analysis approaches and explore different methodological options.

Error handling and robustness represent critical considerations in AIgenerated mathematical code. AI systems can be instructed to include

Verification and testing iterative refinement Integration documentation Performance optimization

comprehensive error checking and validation

comprehensive error checking, input validation, and graceful handling of edge cases. This attention to robustness is particularly important in research applications where code may be used with diverse datasets or parameter ranges.

Example: AI-Generated Numerical Integration Code

Prompt: Generate Python code to implement adaptive quadrature for numerical integration with error estimation.

AI-Generated Code:

```
import numpy as np
   from scipy.integrate import quad
  import matplotlib.pyplot as plt
3
4
   def adaptive_quadrature(f, a, b, tolerance=1e-6, max_depth=10):
5
       . . . .
       Adaptive quadrature implementation with error estimation
7
       .....
       def simpson_rule(f, a, b):
           c = (a + b) / 2
           return (b - a) / 6 * (f(a) + 4*f(c) + f(b))
       def adaptive_recursive(f, a, b, tolerance, depth):
           if depth > max_depth:
               return simpson_rule(f, a, b), False
           c = (a + b) / 2
           whole = simpson_rule(f, a, b)
           left = simpson_rule(f, a, c)
           right = simpson_rule(f, c, b)
           if abs(left + right - whole) < 15 * tolerance:</pre>
               return left + right, True
           else:
               left_result, left_success = adaptive_recursive(
                   f, a, c, tolerance/2, depth+1)
               right_result, right_success = adaptive_recursive(
                   f, c, b, tolerance/2, depth+1)
               return left_result + right_result, left_success and
                     right_success
       result, success = adaptive_recursive(f, a, b, tolerance, 0)
       return result, success
  # Example usage and verification
   def test_function(x):
85
       return np.sin(x) * np.exp(-x)
36
87
  # Test the implementation
   result, success = adaptive_quadrature(test_function, 0, np.pi)
   reference = quad(test_function, 0, np.pi)[0]
```

```
print(f"Adaptive result: {result:.8f}")
print(f"Reference result: {reference:.8f}")
print(f"Error: {abs(result - reference):.2e}")
print(f"Convergence: {success}")
```

Example: AI-Generated Numerical Integration Code weak Goldbach conjecture

Example 10.1 (Numerical Simulation). Let's test the weak Goldbach conjecture, which states that every odd number greater than 5 can be expressed as the sum of three primes.

Prompt to AI

Write a Python script using SymPy to verify the weak Goldbach conjecture for all odd numbers between 7 and 1000.

Al Output

```
import sympy
   def verify_weak_goldbach(limit):
       primes = list(sympy.primerange(2, limit))
       for n in range(7, limit + 1, 2):
           found_sum = False
           for p1 in primes:
               if p1 >= n: break
               for p2 in primes:
                   if p1 + p2 >= n: break
                   for p3 in primes:
                       if p1 + p2 + p3 > n: break
                       if p1 + p2 + p3 == n:
                           found_sum = True; break
                   if found_sum: break
               if found_sum: break
           if not found_sum:
               print(f"Conjecture failed for n = {n}")
               return False
       print("\nConjecture holds for all tested odd numbers.")
       return True
   if ___name___ == '___main___':
23
       verify_weak_goldbach(1000)
```

Prompt to AI

My Python code for prime factorization is slow and sometimes
gives errors. Can you fix and optimize it?
Example 10.2 (Debugging and Refactoring Code). def factor(n):
factors = []
d = 2
while n > 1
while n % d = 0
factors.add(d)
d = d + 1
return factors

Al Output

The provided code has syntax errors and is inefficient. Here is a corrected and optimized version:

```
import math
def prime_factorization(n):
factors = []
while n % 2 == 0:
factors.append(2)
n //= 2
for i in range(3, int(math.sqrt(n)) + 1, 2):
while n % i == 0:
factors.append(i)
factors.append(i)
n //= i
if n > 2:
factors.append(n)
return factors
```

The generation of interactive computational tools represents an exciting frontier in AI-assisted mathematical computing. AI systems can create web-based applications, interactive notebooks, and graphical user interfaces that make mathematical tools accessible to broader audiences. This capability supports both research collaboration and educational applications.

11_Verification and Validation Results

Proof is an idol before whom the pure mathematician tortures himself. — Arthur Eddington (1882–1944)

Science has no national boundaries, but scientists have their motherland. 科学无国界,但科学家有祖国。

— Tsung-Dao Lee (李政道) (1926–2024)

THE VERIFICATION AND VALIDATION OF AI-GENERATED mathematical content represents one of the most critical aspects of responsible AI integration in mathematical research. While AI systems demonstrate impressive capabilities in generating mathematical solutions, proofs, and analyses, the fundamental requirement for mathematical accuracy and rigor means that all AI-generated content must be subjected to careful verification before being accepted or used in research contexts.

The challenge of verification is compounded by the sophisticated nature of modern AI systems, which can produce results that appear mathematically sound and are presented with appropriate mathematical language and formatting. This superficial plausibility can mask underlying errors, logical gaps, or conceptual misunderstandings that require expert analysis to identify. The verification process must therefore go beyond surface-level checking to include deep analysis of mathematical logic, consistency, and correctness.

Systematic verification approaches for AI-generated mathematical content should incorporate multiple layers of checking, each designed to identify different types of potential errors. The first layer involves basic consistency checking, ensuring that mathematical notation is used correctly, that definitions are applied consistently, and that basic algebraic or logical manipulations are performed accurately. This level of verification can often be performed relatively quickly and can identify obvious errors or inconsistencies.

The second layer of verification involves logical structure analysis, examining the overall argument structure of proofs or solutions to ensure that conclusions follow from premises and that each step in the reasoning process is justified. This level of verification requires deeper mathematical understanding and may involve reconstructing key steps of the argument to ensure their validity.

The third layer involves conceptual verification, ensuring that the AI's understanding of mathematical concepts is accurate and that the application of these concepts is appropriate for the given context. This level of verification may require consulting authoritative sources, comparing with known results, or testing the AI's reasoning against established mathematical principles.

basic consistency checking

logical structure analysis

conceptual verification

Independent verification

Independent verification using alternative methods represents a powerful approach to validating AI-generated results. This might involve solving the same problem using different mathematical techniques, implementing independent computational verification, or consulting multiple AI systems to compare their approaches and results. Consistency across different verification methods provides strong evidence for the correctness of results.

The integration of formal verification tools with AI-generated content offers promising possibilities for ensuring mathematical rigor. Proof assistants and formal verification systems can be used to check the logical validity of AI-generated proofs, while computer algebra systems can verify computational results. This hybrid approach combines the natural language capabilities of AI with the rigorous verification capabilities of formal systems.

Testing and validation protocols should be established for different types of AI-generated mathematical content. For computational results, this might involve testing against known solutions, checking limiting cases, or performing sensitivity analysis. For theoretical results, validation might involve checking consistency with established theory, testing implications, or seeking counterexamples.

The documentation of verification processes is crucial for maintaining scientific integrity and enabling reproducibility. This documentation should include not only the verification methods used but also any errors or issues identified during the verification process and how they were addressed. This transparency supports the broader scientific community in understanding and evaluating AI-assisted research.

Collaborative verification approaches leverage the expertise of multiple researchers to ensure the accuracy of AI-generated content. This might involve peer review processes specifically designed for AI-assisted work, collaborative checking of complex proofs or analyses, or the development of community standards for AI verification in mathematical research.

The temporal aspect of verification deserves special attention, as the capabilities and limitations of AI systems evolve rapidly. Verification standards and methods must be regularly updated to address new types of AI capabilities and potential failure modes. This ongoing evolution requires the mathematical community to maintain awareness of AI developments and adapt verification practices accordingly. formal verification tools

Testing and validation protocols

documentation

Collaborative verification

Common Verification Pitfalls

- Overconfidence in AI output: Assuming that sophisticated presentation implies correctness
- **Insufficient testing:** Failing to test edge cases or limiting behaviors
- **Circular verification:** Using AI to verify AI-generated results without independent checks
- **Neglecting conceptual errors:** Focusing on computational accuracy while missing conceptual mistakes
- Inadequate documentation: Failing to record verification processes and findings

The development of automated verification tools specifically designed for AI-generated mathematical content represents an important research direction. These tools might include specialized checking algorithms, pattern recognition systems for identifying common AI errors, and integration platforms that connect AI generation with formal verification systems.

Quality metrics and assessment frameworks for AI-generated mathematical content can help standardize verification processes and provide objective measures of reliability. These metrics might include accuracy rates, consistency measures, and assessments of mathematical rigor that can be applied across different types of mathematical content.

The training of researchers in verification techniques represents a crucial educational component. As AI tools become more prevalent in mathematical research, the ability to effectively verify AI-generated content becomes an essential skill for maintaining scientific integrity. This training should cover both technical verification methods and the development of appropriate skepticism and critical thinking skills. *Common Verification Pitfalls* automated verification tools

Verify Claims

You must independently verify every single claim, calculation, and line of code an AI produces.

The most reliable way to verify a mathematical calculation generated by an LLM is to perform it yourself or, for complex tasks, to use a trusted Computer Algebra System.

Example 11.1 (Symbolic Integration). An LLM can quickly generate the answer to a definite integral, which is much faster and less errorprone than performing integration by parts twice by hand. However, you must validate this result. Verify Claims

Symbolic integration

Prompt to Al

Using SymPy syntax, compute the definite integral of $x^2 \sin(x)$ from 0 to π .

Al Output

Here, the validation is running the trusted SymPy code yourself. SymPy is a CAS, so its result can be relied upon. The LLM's role was to generate the correct code syntax.

Visualizations

A picture is worth a thousand words. An interface is worth a thousand pictures. — **Ben Shneiderman** (1947–)

Mathematics is the foundation of all sciences. 数学是一切科学的基础。

— Su Buqing (苏步青) (1902–2003)

12.1 INTEGRATION OF AI WITH MATHEMATICAL VISUALIZATION TOOLS

THE CREATION OF HIGH-QUALITY MATHEMATICAL GRAPHICS AND VI-SUALIZATIONS represents a crucial aspect of mathematical research and communication, and AI tools are increasingly capable of assisting in this process. Modern AI systems can generate code for creating sophisticated visualizations, suggest appropriate graphical representations for mathematical concepts, and even assist in the design of publication-quality figures that meet professional standards.

The integration of AI with mathematical visualization tools creates powerful synergies for creating both explanatory and research-oriented graphics. AI systems can translate conceptual descriptions of desired visualizations into executable code, suggest appropriate plotting techniques for different types of mathematical data, and generate customized visualization solutions that address specific research needs. This capability is particularly valuable for researchers who may have clear ideas about what they want to visualize but lack the technical expertise to implement complex graphics.

Python has emerged as a particularly powerful platform for AI-assisted mathematical visualization due to its extensive ecosystem of plotting libraries and the fluency of AI systems in generating Python code. Libraries such as Matplotlib, Plotly, and Seaborn provide comprehensive capabilities for creating mathematical plots, while specialized libraries like Mayavi and VTK enable three-dimensional visualization of mathematical objects and data.

The generation of publication-quality graphics requires attention to numerous technical details that AI systems can help manage. This includes proper handling of mathematical notation in labels and titles, appropriate scaling and aspect ratios for different types of plots, consistent color schemes and styling, and optimization for different output formats. AI systems can generate code that addresses these technical requirements Python

publication-quality graphics



while allowing researchers to focus on the mathematical content of their visualizations.

Matplotlib, as the foundational plotting library in Python, provides extensive capabilities for creating mathematical graphics, and AI systems demonstrate strong proficiency in generating Matplotlib code. The library's integration with LATEX for mathematical typesetting makes it particularly suitable for creating publication-quality figures that include complex mathematical notation. AI-generated Matplotlib code can handle everything from simple function plots to complex multi-panel figures with sophisticated annotation and styling.

The creation of interactive visualizations represents an exciting frontier in AI-assisted mathematical graphics. Tools like Plotly and Bokeh enable the creation of interactive plots that allow users to explore mathematical relationships dynamically. AI systems can generate code for creating these interactive visualizations, including user interface elements, responsive behaviors, and dynamic updates based on user input.

Three-dimensional visualization of mathematical objects presents unique challenges and opportunities for AI assistance. Libraries like Mayavi and VTK provide powerful capabilities for creating three-dimensional mathematical visualizations, but they require significant technical expertise to use effectively. AI systems can generate code that leverages these libraries to create sophisticated three-dimensional plots, surface visualizations, and animated mathematical objects.

The integration of symbolic mathematics with visualization creates powerful possibilities for mathematical exploration. AI systems can generate code that combines symbolic computation libraries like SymPy with visualization tools to create graphics that automatically update based on mathematical relationships. This integration enables the creation of dynamic visualizations that respond to changes in mathematical parameters or expressions.

Animation and dynamic visualization represent important capabilities for illustrating mathematical concepts and processes. AI systems can generate code for creating mathematical animations that show the evolution of mathematical objects over time, the behavior of iterative processes, or the relationships between different mathematical parameters. These animations can be particularly valuable for educational applications and for illustrating complex mathematical concepts.

The optimization of graphics for different output formats requires careful attention to technical details. Vector formats like PDF and SVG are generally preferred for publication due to their scalability and crisp rendering of mathematical notation, while raster formats may be more appropriate for web applications or presentations. AI systems can generate code that optimizes graphics for specific output requirements and applications.

Matplotlib interactive visualizations Three-dimensional visualization symbolic mathematics dynamic visualization optimization of graphics
Example: AI-Generated 3D Mathematical Visualization

Prompt: Create a 3D visualization of the function $f(x, y) = \sin(x^2 + y^2)e^{-(x^2+y^2)/4}$ with contour lines and gradient field.

AI-Generated Code:

```
<sup>1</sup> import numpy as np
<sup>2</sup> import matplotlib.pyplot as plt
3 from mpl_toolkits.mplot3d import Axes3D
   from matplotlib import cm
4
   import matplotlib.patches as patches
   # Define the function
7
   def f(x, y):
8
        r_2 = x * * 2 + y * * 2
9
        return np. sin (r2) * np. exp(-r2/4)
10
12 # Create meshgrid
x_3 = np.linspace(-3, 3, 100)
   y = np.linspace(-3, 3, 100)
14
   X, Y = np.meshgrid(x, y)
15
  Z = f(X, Y)
  # Create figure with subplots
19
  fig = plt.figure(figsize=(15, 5))
21 # 3D surface plot
22 ax1 = fig.add_subplot(131, projection='3d')
23 surf = ax1.plot_surface(X, Y, Z, cmap=cm.viridis, alpha=0.8)
  ax1.set_title(r'f(x,y) = \frac{x^2+y^2}{e^{-(x^2+y^2)/4}}
24
25 ax1.set_xlabel('x')
26 ax1.set_ylabel('y')
27 ax1.set_zlabel('f()
  ax1.set_zlabel('f(x,y)')
  fig.colorbar(surf, ax=ax1, shrink=0.5)
  # Contour plot
30
ax_2 = fig.add_subplot(132)
  contour = ax2.contour(X, Y, Z, levels=20, cmap=cm.viridis)
32
ax2.clabel(contour, inline=True, fontsize=8)
34 ax2.set_title('Contour Lines')
35 ax2.set_xlabel('x')
  ax2.set_ylabel('y')
36
37
  ax2.set_aspect('equal')
38
39 # Gradient field
  ax_3 = fig.add_subplot(133)
40
  # Calculate gradient
41
dx = np.gradient(Z, axis=1)
dy = np.gradient(Z, axis=0)
  # Subsample for clarity
44
  skip = 8
45
   ax3.quiver(X[::skip, ::skip], Y[::skip, ::skip],
               dx[::skip, ::skip], dy[::skip, ::skip],
47
  scale=20, alpha=0.7)
ax3.contour(X, Y, Z, levels=10, alpha=0.3, colors='gray')
48
49
   ax3.set_title('Gradient Field')
50
  ax3.set_xlabel('x')
51
  ax3.set_ylabel('y')
52
  ax3.set_aspect('equal')
53
54
  plt.tight_layout()
55
```

plt.show()

Example 12.1 (Generating Plots with Matplotlib). Here is an example using Matplotlib. Matplotlib is a Python library and provides a MATLAB-like plotting interface through its pyplot module.

Prompt to AI

Write a Python script using Matplotlib to plot the function $f(x) = \frac{\sin(\pi x)}{\pi x}$ and its envelope functions $g(x) = \pm \frac{1}{\pi x}$ for $x \in [-5, 5]$. Label the axes and add a title.

AI Output

```
import numpy as np
import numpy as np
import matplotlib.pyplot as plt

X = np.linspace(-5, 5, 1000)
X = x[x != 0]

public func = np.sin(np.pi * x) / (np.pi * x)
envelope = 1 / (np.pi * np.abs(x))

public figure(figsize=(10, 6))
public plot(x, sinc_func, label=r'$f(x) = \frac{\sin(\pi x)}{\pi x}
}')

public plot(x, envelope, 'r--', label=r'Envelope $\pm \frac{1}{|\pi x|}$')

public plot(x, -envelope, 'r--')
public plot(x
```

The integration of AI-generated graphics with LATEX document preparation systems represents an important workflow consideration. AI systems can generate code that produces graphics in formats that integrate seamlessly with LATEX, including proper handling of fonts, sizing, and mathematical notation. This integration is crucial for maintaining consistency between textual and graphical elements in mathematical documents.

Color theory and accessibility considerations in mathematical graphics represent important aspects that AI systems can help address. This includes generating graphics that are accessible to individuals with color vision deficiencies, ensuring sufficient contrast for readability, and selecting color schemes that are appropriate for the intended audience and publication medium.

The documentation and reproducibility of AI-generated graphics require special attention. This includes maintaining records of the code used to generate graphics, documenting any manual modifications or enhance*Example: AI-Generated 3D Mathematical Visualization* Matplotlib

Color theory and accessibility

documentation and reproducibility

ments, and ensuring that graphics can be regenerated if needed. This documentation is crucial for maintaining scientific integrity and enabling collaboration.

Quality control and review processes for AI-generated graphics should include verification of mathematical accuracy, assessment of visual clarity and effectiveness, and checking for consistency with publication standards. This review process should involve both technical verification of the underlying mathematics and aesthetic evaluation of the visual presentation.

For papers and posters, always use vector graphics (.pdf, .svg, .eps) over raster graphics (.png, .jpg). Vector graphics are mathematical descriptions of shapes and are infinitely scalable, ensuring they are always crisp.

12.2 The Art and Science of Mathematical Graphics

A clear, well-designed graphic can illuminate a complex idea more effectively than pages of text. A poor graphic can confuse and mislead.

Figure 12.1 demonstrates the stark difference between effective and ineffective mathematical graphics. The poor example exhibits several common problems: excessive grid lines that distract from the data, garish colors that are difficult to distinguish and unprofessional in appearance, and text that is too small to read clearly. In contrast, the good example shows how professional mathematical graphics should appear: clean lines, a limited color palette that maintains readability, and appropriately sized text with clear axis labels. The comparison of vector versus raster graphics at the bottom illustrates why vector formats (.pdf, .svg, .eps) are essential for academic publications—they remain crisp at any scale, unlike raster images that become pixelated when enlarged.



Figure 12.1: Comparison of poor and good mathematical graphics practices. The poor example shows common mistakes including cluttered elements, inappropriate colors, and unreadable text. The good example demonstrates professional formatting with clear labels, appropriate color schemes, and readable fonts. The bottom panels illustrate the critical difference between raster and vector graphics for academic publications.

Quality control and review

Principles of Effective Visualization

- **Clarity Over Clutter:** Every element should serve a purpose. Avoid distracting backgrounds, 3D effects, and unnecessary colors ("chartjunk").
- Label Everything: Axes, titles, legends, and important data points should be clearly and legibly labeled. Use font sizes that will be readable when the figure is scaled down for a paper.
- Use Color Purposefully: Use color to distinguish categories of data, not for decoration. Be mindful of colorblindness; use palettes that are distinguishable in grayscale.
- Choose the Right Format: This is the most crucial technical decision.

Vector vs. Raster: A Critical Distinction

There are two families of image formats. Understanding the difference is non-negotiable for academic publishing.

- **Raster Graphics** (or bitmap) store an image as a grid of pixels. Examples: .png, .jpg, .gif. They are ideal for photographs but scale poorly. When you zoom in, you see the individual pixels, resulting in a "blocky" or "blurry" look.
- Vector Graphics store an image as a set of mathematical instructions (e.g., "draw a line from point A to point B"). Examples: .pdf, .svg, .eps. They are infinitely scalable. A vector graphic will remain perfectly crisp and clear no matter how much you zoom in.

For papers and posters, always use vector formats for plots and diagrams.

Exporting from Your Tools

• Python/Matplotlib: When saving, specify a vector format.

```
1 # Instead of plt.show(), use plt.savefig()
2 plt.savefig("goldbach_comet.pdf", format="pdf", bbox_inches="tight"
        )
3 plt.savefig("goldbach_comet.svg", format="svg", bbox_inches="tight"
        )
```

• Mathematica: Use the 'Export' command.

```
1 (* Assuming plotObject is a graphic you've generated *)
2 Export["damped_wave.pdf", plotObject]
3 Export["damped_wave.svg", plotObject]
```

• IAT_EX/TikZ: This is the purest vector format, as it is compiled directly with your document. The diagrams in this guide were made with TikZ. It has a steep learning curve but offers unparalleled quality and control. Use LLMs to generate initial TikZ code!



Figure 12.2: A visual comparison of vector and raster graphics when scaled.

Example 12.2 (Generating Diagrams with TikZ). PGF/TikZ is a pair of languages for producing vector graphics from a geometric/al-gebraic description. This can be embedded in a LATEX document to generate diagrams, plots, and other visualizations.

LATEX code using TikZ

Prompt to AI

Generate LATEX code using the TikZ package to draw a cycle graph C_4 with labeled vertices.

Al Output

12.3 Refining Graphics

WHILE GENERATING GRAPHICS FROM SCRATCH IS A POWERFUL CA-PABILITY, one of the most common and effective uses of AI is in *refining* existing code. Often, the first version of a graphic—whether written by a human or generated by an AI—is just a starting point. AI can act as an expert assistant to improve aesthetics, add complexity, and increase the modularity of your code.

Example 12.3 (Refining a TikZ Diagram). Suppose we used an AI to generate a simple diagram of a cycle graph C_4 , and it produced the following functional but basic code and the corresponding image shown in Figure 12.3 (left).

```
% Initial, basic TikZ code for C4
\begin{tikzpicture}
    \node[circle,draw,fill=blue!20] (v1) at (0,2) {$v_1$};
    \node[circle,draw,fill=blue!20] (v2) at (2,0) {$v_2$};
```

```
\node[circle,draw,fill=blue!20] (v3) at (0,-2) {$v_3$};
\node[circle,draw,fill=blue!20] (v4) at (-2,0) {$v_4$};
\draw[thick] (v1) -- (v2) -- (v3) -- (v4) -- cycle;
\end{tikzpicture}
```

The diagram is correct, but the code is repetitive and not easy to modify. Now, we want to adapt this diagram to communicate a more specific mathematical idea: highlighting the path graph P_3 as a subgraph within C_4 . This requires more than just a new diagram; it requires a more sophisticated code structure. We can ask the AI to perform this refinement.

Prompt to AI

I have this basic TikZ code for a cycle graph C_4 . I need you to refine it to highlight the path P_3 (from v_1 to v_2 to v_3) as a subgraph. Please do the following: 1. Restructure the code to use \tikzset or local /.style definitions for nodes and edges to make it more modular. 2. Create separate styles for "standard" elements (in gray) and "highlighted" elements (in red). 3. Apply the highlight style to nodes v_1 , v_2 , v_3 and the edges connecting them. 4. Make the highlighted edges thicker to make them stand out.

Al Output

Here is the refined TikZ code that uses modular styles to highlight the specified subgraph:

```
\begin{tikzpicture}[
```

```
% Define reusable styles
```

node_std/.style={circle, draw=gray, fill=gray!20, thick,





Figure 12.3: A comparison of the initial AI-generated TikZ diagram (left) and the refined version after a specific modification prompt (right). The refined version uses modular styles and highlights a specific mathematical substructure.

Initial AI-Generated Graphic Refined Graphic This iterative process transformed a simple, correct diagram into a more sophisticated and communicative one. The refined code is not only aesthetically better but also more maintainable and adaptable for future use, demonstrating the value of using AI as a partner in the creative process.



The best way to communicate is to write clearly.

— Donald Knuth (1938–)

Mathematics is beautiful, and it should be presented beautifully. 数学是美的,应该美地呈现出来。

— **Shiing-Shen Chern** (陈省身) (1911 – 2004)

THE INTEGRATION OF AI TOOLS WITH LATEX DOCUMENT PREPARATION represents a transformative development in mathematical writing and publishing. AI systems can assist with virtually every aspect of LATEX document creation, from initial document structure and formatting to complex mathematical typesetting and bibliography management. This integration enables researchers to focus on mathematical content while leveraging AI capabilities to handle the technical aspects of document preparation.

Modern AI systems demonstrate remarkable proficiency in LATEX syntax and can generate properly formatted mathematical documents from natural language descriptions. This capability includes not only basic document structure but also sophisticated mathematical typesetting, complex table and figure formatting, and integration with specialized LATEX packages for mathematical applications. The ability to describe desired formatting in natural language and receive working LATEX code significantly reduces the barrier to entry for professional mathematical typesetting.

The generation of mathematical content in LATEX format represents one of the most valuable applications of AI integration. AI systems can convert mathematical expressions from various formats into properly formatted LATEX code, generate complete mathematical derivations with appropriate formatting, and create complex mathematical displays that follow professional typesetting standards. This capability is particularly valuable for researchers who may be comfortable with mathematical concepts but less familiar with LATEX mathematical typesetting conventions.

Document structure and organization represent areas where AI can provide significant assistance. AI systems can generate LATEX document templates that are appropriate for different types of mathematical publications, suggest organizational structures for complex documents, and create consistent formatting schemes that enhance document readability. This structural assistance is particularly valuable for lengthy documents such as theses, research papers, and technical reports.

The integration of AI with specialized LATEX packages opens numerous possibilities for enhanced mathematical typesetting. Packages such as amsmath, amsthm, and amssymb provide extensive capabilities for mathematical formatting, while packages like tikz and pgfplots enable the creation of sophisticated mathematical diagrams and plots directly within Document structure and organization

integrating packages



LATEX documents. AI systems can generate code that leverages these packages effectively, creating professional-quality mathematical documents with minimal manual intervention.

Bibliography management represents another area where AI can provide valuable assistance. AI systems can generate BibTeX entries from various citation formats, suggest appropriate citation styles for different publication venues, and create formatted bibliographies that meet professional standards. This capability is particularly valuable for managing large numbers of references and ensuring consistent citation formatting throughout lengthy documents.

The creation of mathematical presentations using Beamer represents an important application of AI-assisted LATEX document preparation. AI systems can generate Beamer presentations with appropriate mathematical formatting, suggest effective slide layouts for different types of mathematical content, and create consistent visual themes that enhance presentation effectiveness. This capability enables researchers to create professional mathematical presentations with minimal investment in learning complex Beamer syntax.

Table and figure formatting in IATEX documents can be significantly enhanced through AI assistance. AI systems can generate properly formatted tables for mathematical data, create figure environments with appropriate captions and references, and handle complex multi-panel figures with consistent formatting. This capability is particularly valuable for documents that include extensive numerical results or multiple graphical elements.

The automation of repetitive LATEX formatting tasks represents a significant productivity enhancement. AI systems can generate consistent formatting for mathematical definitions, theorems, and proofs, create standardized section headers and document elements, and apply consistent styling throughout lengthy documents. This automation reduces the time required for document preparation and minimizes formatting errors.

Cross-referencing and document navigation represent important aspects of professional mathematical documents that AI can help manage. AI systems can generate appropriate label and reference schemes, create consistent numbering systems for mathematical objects, and implement navigation aids such as hyperlinks and bookmarks. This capability is particularly important for complex documents with extensive internal cross-references.

Example: AI-Generated LATEX Theorem Environment

Prompt: Create a LAT_EX document section with a theorem about continuity, its proof, and a related example.

AI-Generated LATEX:

- \section {Continuity and Limits}
- \begin{theorem}[Epsilon-Delta Characterization of Continuity]

Bibliography management

presentations

formatting

Cross-referencing and navigation

```
\label{thm:continuity_epsilon_delta}
    Let f: \mathbb{R} \setminus f \in \mathbb{R} \setminus f.
Then f \in \mathbb{R} \setminus f and f \in \mathbb{R} \setminus f.
Then f \in \mathbb{R} \setminus f is continuous at f \in f and only if for every \mathbb{R} \setminus f and f \in \mathbb{R} \setminus f.
Then f \in \mathbb{R} \setminus f is continuous at f \in \mathbb{R} \setminus f.
Then f \in f is continuous at f \in \mathbb{R} \setminus f.
     1/
     |x - a| < |delta | (x) - f(a)| < |varepsilon|
     \mathbb{N}^{1}
    \end{ theorem }
    \begin { proof }
    We prove both directions of the equivalence.
     \textbf{($\Rightarrow$)} Assume $f$ is continuous at $a$.
    By definition of continuity, \Pi_{x \to a} f(x) = f(a).
Let \nabla_{x \to a} f(x) = f(a).
there exists \delta = 0 be given. By the definition of limit,
there exists \delta = 0 such that for all x \in a:
     |x - a| < |delta | (x) - f(a)| < |varepsilon|
23
    Since this inequality also holds trivially when x = a
     (as \final f(a) - f(a) = o < \mathbf{varepsilon}), we have the
     desired result.
    \textbf{($\Leftarrow$)} Assume the epsilon-delta condition holds.
    We need to show that \lim_{x \to a} f(x) = f(a).
29
    Let varepsilon > 0 be given. By hypothesis, there exists delta > 0 such that |x - a| < delta implies |f(x) - f(a)| < varepsilon. This is precisely the
     definition of \lim_{x \to a} f(x) = f(a).
    \end{proof}
     \begin {example}
    (habel {ex: quadratic_continuity})
Consider f(x) = x^2. We show that ff is continuous at
    a = 2 using the epsilon-delta definition.
    Let varepsilon > 0 be given. We need to find delta > 0 such that |x - 2| < delta implies |x^2 - 4| < varepsilon.
    Note that:
     \begin { align }
     |x^2 - 4| \& = |x - 2| |x + 2|
    \end{align}
    If we restrict |x - 2| < 1, then 1 < x < 3, so
    |x + 2| < 5. Thus:
     1/
    |x^2 - 4| = |x - 2||x + 2| < 5|x - 2|
     \mathbf{M}
    Choose \delta = \min \left\{1, \frac{1}{\sqrt{1}}\right\}.
    Then for |x - 2| < \ delta:
     1/
     |x^2 - 4| < 5|x - 2| < 5 \setminus cdot \setminus frac \{ \setminus varepsilon \} \{5\} = \setminus varepsilon \}
     \mathbf{N}
     Therefore, f(x) = x^2 is continuous at x = 2.
     \end{example}
```

Result: AI-Generated LATEX Theorem Environment

10.1 CONTINUITY AND LIMITS

Theorem 13.1 (Epsilon-Delta Characterization of Continuity). Let f: $\mathbb{R} \to \mathbb{R}$ and let $a \in \mathbb{R}$. Then f is continuous at a if and only if for every $\varepsilon > 0$, there exists $\delta > 0$ such that for all $x \in \mathbb{R}$:

 $|x-a| < \delta \Rightarrow |f(x) - f(a)| < \varepsilon$

Proof. We prove both directions of the equivalence. (\Rightarrow) Assume *f* is continuous at *a*. By definition of continuity, $\lim_{x\to a} f(x) = f(a)$. Let $\varepsilon > 0$ be given. By the definition of limit, there exists $\delta > 0$ such that for all $x \neq a$: $|x-a| < \delta \Rightarrow |f(x) - f(a)| < \varepsilon$ Since this inequality also holds trivially when x = a (as |f(a) - a| $|f(a)| = 0 < \varepsilon$, we have the desired result. (⇐) Assume the epsilon-delta condition holds. We need to show that $\lim_{x\to a} f(x) = f(a)$. Let $\varepsilon > 0$ be given. By hypothesis, there exists $\delta > 0$ such that $|x - a| < \delta$ implies $|f(x) - f(a)| < \varepsilon$. This is precisely the definition of $\lim_{x\to a} f(x) = f(a)$. **Example 13.1.** Consider $f(x) = x^2$. We show that *f* is continuous at a = 2 using the epsilon-delta definition. Let $\varepsilon > 0$ be given. We need to find $\delta > 0$ such that $|x - 2| < \delta$ implies $|x^2 - 4| < \varepsilon$. Note that: $|x^2 - 4| = |x - 2||x + 2|$ If we restrict |x - 2| < 1, then 1 < x < 3, so |x + 2| < 5. Thus: $|x^{2}-4| = |x-2||x+2| < 5|x-2|$ Choose $\delta = \min \{1, \frac{\varepsilon}{5}\}$. Then for $|x - 2| < \delta$: $|x^2-4| < 5|x-2| < 5 \cdot \frac{\varepsilon}{5} = \varepsilon$

Therefore, $f(x) = x^2$ is continuous at x = 2.

The integration of AI-generated content with existing LATEX workflows requires attention to consistency and compatibility. AI systems can generate LATEX code that integrates seamlessly with existing document structures, maintains consistent formatting styles, and respects established conventions for mathematical typesetting. This integration capability is crucial for collaborative projects and for maintaining consistency across multiple documents.

Error handling and debugging in AI-generated LATEX code represent important considerations for reliable document preparation. AI systems can be instructed to generate robust LATEX code that includes appropriate error checking, handles edge cases gracefully, and provides clear error messages when problems occur. This attention to robustness is particularly important for complex documents that may undergo multiple revisions and modifications.

The customization and adaptation of AI-generated LATEX code enables researchers to tailor document formatting to specific requirements. AI systems can modify generated code to meet particular journal specifications,

Result: AI-Generated LATEX Theorem Environment

Error handling and debugging

customization and adaptation

adapt formatting for different document types, and implement custom styling requirements. This flexibility ensures that AI-generated LATEX code can be adapted to diverse publication contexts and institutional requirements.

Version control and collaboration considerations become particularly important when integrating AI-generated LATEX code into collaborative research projects. This includes maintaining clear records of AI-generated content, documenting modifications and customizations, and ensuring that collaborative workflows can accommodate AI-assisted document preparation. These practices support effective collaboration while maintaining transparency about the role of AI in document creation.

AI can be an invaluable partner when writing LATEX documents, especially for complex syntax.

Example 13.2 (Generating Document Templates). LLMs can generate results and display them in a LATEX document.

Prompt to Al

Create a basic LATEX article template for a math paper. Include packages for math, theorems, graphics, and a bibliography.

Al Output

The preamble of this very document is a perfect example of such a generated template. It includes packages for geometry, fonts, math ('amsmath', 'amsthm'), graphics ('graphicx', 'tikz', 'xcolor'), code listings, and bibliography ('biblatex').

Example 13.3 (Typesetting Complex Mathematics). LLMs can produce complicated LAT_FX code.

Prompt to AI

How do I typeset the following in LATEX: The set of integers \mathbb{Z} , a 2x2 matrix with entries a, b, c, d, and the definition of the limit of a sequence x_n .

Al Output

Here are the LATEX commands for each:

- The set of integers: 'Z'. Requires \usepackage{amssymb}.
- A 2×2 matrix:

 $\left(\begin{array}{cc}a&b\\c&d\end{array}\right).$

Requires \usepackage{amsmath}.

• Limit definition: $\forall \epsilon > 0, \exists N \in \mathbb{N}$ such that $n > N \Rightarrow |x_n - L| < \epsilon$.

Version control and collaboration

Document Templates

Presentations and Beamer

In the realm of mathematics, there is no place for authority; only truth matters. If you can't explain it simply, you don't understand it well enough.

— Albert Einstein (1879–1955)

In the realm of mathematics, there is no place for authority; only truth matters. 在数学的领域里,没有权威的地位,只有真理是重要的。

— Chen Jingrun (陈景润) (1933–1996)

THE CREATION OF PROFESSIONAL MATHEMATICAL PRESENTATIONS using LATEX Beamer represents an important application of AI-assisted document preparation. Modern AI systems can generate complete Beamer presentations with appropriate mathematical formatting, suggest effective slide layouts for different types of mathematical content, and create consistent visual themes that enhance presentation effectiveness. This capability enables researchers to create high-quality mathematical presentations with minimal investment in learning complex Beamer syntax.

AI systems demonstrate particular strength in generating Beamer presentations that effectively communicate mathematical concepts. This includes creating slide structures that support logical flow of mathematical arguments, generating appropriately formatted mathematical displays that are readable in presentation contexts, and suggesting visual elements that enhance comprehension of complex mathematical ideas. The ability to describe presentation goals in natural language and receive working Beamer code significantly reduces the barrier to creating professional mathematical presentations.

The integration of mathematical content with visual presentation elements represents a crucial aspect of effective mathematical presentations. AI systems can generate Beamer code that balances mathematical rigor with visual clarity, creating presentations that are both mathematically accurate and accessible to diverse audiences. This includes appropriate use of mathematical notation, effective spacing and layout of mathematical expressions, and integration of graphical elements that support mathematical exposition.

Slide design and layout optimization for mathematical presentations require careful consideration of readability, visual hierarchy, and audience engagement. AI systems can generate Beamer templates that are optimized for mathematical content, suggest appropriate font sizes and spacing for different presentation contexts, and create consistent visual themes that enhance professional appearance. This design assistance is particularly valuable for researchers who may be experts in mathematical content but less familiar with presentation design principles.

The creation of animated mathematical presentations represents an

advanced application of AI-assisted Beamer generation. AI systems can generate code for creating mathematical animations that illustrate dynamic processes, show the evolution of mathematical proofs, or demonstrate the behavior of mathematical systems over time. These animated presentations can be particularly effective for explaining complex mathematical concepts and engaging audiences with interactive content.

Example: AI-Generated Beamer Presentation on Fourier Series	
definitions, examples, and visualizations.	
AI-Generated Beamer Code:	
<pre>1 \documentclass[12pt]{beamer} 2 \usetheme{Madrid} 3 \usepackage{amsmath,amssymb} 4 \usepackage{graphicx} 5 \usepackage{tikz} 6 \usepackage{pgfplots} 7 \pgfplotsset{compat=1.16} 8 9 \title{Introduction to Fourier Series} 0 \subtitle{Decomposing Functions into Trigonometric Components} 1 \author{Your Name}</pre>	
2 \institute {Department of Mathematics} 3 \date {\today}	
5 \begin {document} 6 7 \frame{\titlepage}	
 8 9 \begin {frame} 0 \frametitle {Outline} 1 \tableofcontents 2 \end{frame} 33 44 \section {Introduction and Motivation} 	
 begin {frame} \begin {frame} \frametitle {What are Fourier Series?} \begin {block }{Main Idea} Any periodic function can be expressed as a sum of sine and cosine functions with different frequencies. \end{block} 	
22 33 \pause 34	
5 \begin{block}{Mathematical Expression} 6 For a periodic function \$f(x)\$ with period \$2\pi\$: 1 \[
$\begin{cases} f(x) = \frac{a_0}{2} + \frac{a_n}{2} + \frac{a_n}{2} + \frac{a_n}{2} + \frac{a_n}{2} \end{cases}$ $\begin{cases} 1 \\ 9 \\ 1 \\ 9 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\$	
22 33 \pause	
 begin {alertblock }{Key Insight } Complex periodic phenomena can be understood as combinations of simple harmonic components. \end{alertblock} \end{frame} 	
v section {Mathematical Foundation}	
- 3 \begin{frame} 4 \frametitle{Fourier Coefficients} 5 The coefficients are determined by:	
⁶ 77 \begin{align}	

Fourier series

```
a_o &= \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \, dx \\[o.5em]
a_n &= \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(nx) \, dx \\[o.5em]
b_n &= \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin(nx) \, dx
\end{align}
// vend{align}
// vend{align}
// trigonometric functions:
// These formulas arise from the orthogonality of
// trigonometric functions:
// int_{-\pi}^{\pi} \cos(mx) \cos(nx) \, dx =
// begin{cases}
// o & \text{if } m \neq n \\
// \pi & \text{if } m = n \neq o
// \end{block}
// \end{block}
// end{frame}
```

Example: AI-Generated Beamer Presentation (cont'd)

AI-Generated Beamer Code:

1	\section {Example: Square Wave}
2	\hogin {frame}
3	\frametitle [Square Wave Eurotion]
4	\hegin { columns}
5	\column {o r \ toxtwidth }
0	Define the square wave:
7	\[
0	$f(x) = \langle hogin cases \rangle$
9	$1 \left\{ k \right\} = \left\{ begin \left\{ cases \right\} \right\}$
1	$-1 & \operatorname{text} \{ if \} - ni < x < 0$
1	\end{cases}
12	$\langle 1 \rangle$
14	,1
15	\pause
16	
7	Computing the coefficients:
8	\begin { align }
9	a_0 &= 0 \\
20	a_n &= o \\
21	$b_n \&= \frac{1}{n \cdot pi} \cdot text \{ (odd \} n \cdot text \}$
22	\end{align}
23	
24	\column {0.5 \ textwidth }
25	\begin { center }
26	\begin {tikzpicture }[scale = 0.8]
27	\begin { axis }[
28	domain = -3.5:3.5,
29	samples=200,
30	xlabel= [¢] x [¢]
1	$x_{1abe1-\varphi,\varphi}$, $x_{1abe1-\varphi,f}(x)$ φ
12	$v_{min} = -1$
13	ymm = 1.5, ymax = 1.5
1-	grid=maior
16]
17	\addplot[blue, thick, samples=50] {
18	(x > -pi && x < 0)? -1 :
39	(x > 0 & x < pi)? 1 : 0
40	};
41	\end{axis}
42	\end{ tikzpicture }
43	\end{center}
44	\end { columns }
45	\end { frame }
46	
47	\begin {trame}
48	\frametitle {Square Wave Fourier Series}
49	The Fourier series representation:

Example: AI-Generated Beamer Presentation on Fourier Series

Square Wave Frames

```
f(x) = \frac{4}{\sqrt{pi}} \quad sum_{n=1,3,5,\ldots}^{ \min\{y\} \quad \max\{n=1,3,5,\ldots\}}
   \setminus]
3
   \pause
55
   \begin{block}{Partial Sums}
  S_{5}(x) &= \frac{1}{10} \left[ \frac{1}{10} + \frac{1}{10} \right] \\
       sin(5x) {5} \right)
   \end{align}
   \end{block}
   \pause
65
   \begin{alertblock}{Convergence}
   As we add more terms, the series converges to the
67
   square wave function.
   \end{alertblock}
69
   \end{frame}
```

Example: AI-Generated Beamer Presentation (cont'd)

Example: AI-Generated Beamer Presentation (cont'd)

AI-Generated Beamer Code:

```
\section { Applications }
    \begin { frame }
    \frametitle{Applications of Fourier Series}
    \begin {enumerate}
   \item \textbf{Signal Processing}
       \begin{itemize}
       \item Audio compression (MP3)
       \item Image processing (JPEG)
\item Digital communications
       \end{itemize}
3
   \pause
4
   \item \textbf{Physics and Engineering}
       \begin{itemize}
       \item Heat conduction
       \item Vibration analysis
       \item Electrical circuits
       \end{itemize}
22
   \pause
3
   \item \textbf{Mathematics}
       \begin{itemize}
       \item Solving differential equations
       \item Harmonic analysis
       \item Number theory
       \end{itemize}
   \end{enumerate}
0
   \end{frame}
   \begin { frame }
   \frametitle {Conclusion }
   \begin{block}{Key Points}
   \begin{itemize}
   \item Fourier series decompose periodic functions into
         harmonic components
   \item The coefficients are computed using orthogonality
   \item Applications span multiple fields of science and
          engineering
    \end{itemize}
   \end{block}
   \pause
45
```

Fourier Series Applications Frames

```
begin{block}{Next Steps}
begin{itemize}
\item Fourier transforms for non-periodic functions
\item Discrete Fourier Transform (DFT)
\item Fast Fourier Transform (FFT) algorithms
\end{itemize}
\end{frame}
\end{document}
```

Integration with mathematical graphics and visualizations represents another important aspect of AI-assisted Beamer presentations. AI systems can generate code that seamlessly integrates mathematical plots, diagrams, and visualizations into presentation slides, ensuring consistent formatting and appropriate sizing for presentation contexts. This integration capability enables the creation of visually rich mathematical presentations that effectively communicate complex ideas.

The management of mathematical notation and symbols in presentation contexts requires special attention to readability and clarity. AI systems can generate Beamer code that optimizes mathematical notation for presentation display, ensuring that mathematical expressions are clearly readable from typical viewing distances and that symbol sizing and spacing are appropriate for projection environments.

The customization of Beamer themes and styling for mathematical presentations represents an important aspect of professional presentation design. AI systems can generate custom Beamer themes that are optimized for mathematical content, suggest appropriate color schemes and fonts for different presentation contexts, and create consistent visual branding that enhances professional appearance. This customization capability enables researchers to create presentations that align with institutional branding or personal preferences while maintaining readability and professionalism.

Interactive elements and audience engagement represent advanced applications of AI-assisted Beamer presentations. AI systems can generate code for creating interactive mathematical demonstrations, clickable mathematical objects, and dynamic content that responds to audience input. These interactive elements can significantly enhance audience engagement and provide more effective communication of complex mathematical concepts.

The integration of multimedia elements with mathematical content represents another important capability of AI-assisted Beamer generation. AI systems can generate code that seamlessly integrates videos, animations, and interactive elements into mathematical presentations, ensuring that multimedia content enhances rather than detracts from mathematical exposition. This integration capability enables the creation of rich, multimodal presentations that appeal to diverse learning styles.

Accessibility considerations in mathematical presentations require special attention to ensure that content is accessible to individuals with Example: AI-Generated Beamer Presentation (cont'd) graphics and visualizations

mathematical notation and symbols

themes and styles

multimedia elements

Interactive elements

Accessibility

diverse needs and abilities. AI systems can generate Beamer code that incorporates accessibility features such as alternative text for images, highcontrast color schemes, and clear navigation structures. This attention to accessibility ensures that mathematical presentations can reach broader audiences and comply with institutional accessibility requirements.

The optimization of presentations for different delivery contexts represents a practical consideration that AI systems can help address. This includes generating presentations that are optimized for different screen sizes and resolutions, creating handout versions that are suitable for printing, and adapting content for online delivery platforms. This optimization capability ensures that presentations remain effective across diverse delivery contexts.

AI can generate the boilerplate code for presentations using the 'beamer' class, saving you time on setup and allowing you to focus on the content.

Example 14.1 (Creating a Minimal Beamer Presentation). Next ...

Prompt to Al

Generate a minimal Beamer presentation LATEX template with a title slide and one frame containing a theorem and a block.

Al Output

1	\documentclass{beamer}
2	\usetheme{Madrid}
3	<pre>\usepackage{amsmath, amsthm}</pre>
4	
5	<pre>\title{My Research Presentation}</pre>
6	<pre>\author{Your Name}, \institute{Your University}, \date{\today}</pre>
7	
8	<pre>\begin{document}</pre>
9	\frame{\titlepage}
10	
11	\begin{frame}
12	\frametitle{Main Results}
13	<pre>\begin{theorem}[Fermat's Last Theorem]</pre>
14	For any integer $n > 2$, there are no positive integers x , y,
	z \$ such that $x^n + y^n = z^n$ \$.
15	\end{theorem}
16	\begin{block}{Key Idea}
17	The proof by Andrew Wiles connects this problem to the
	modularity theorem for elliptic curves.
18	\end{block}
19	\end{frame}
20	\end{document}

optimize presentations

Minimal Beamer Presentation

Figure 14.1: Introductory slides explaining the concept of Fourier series.

	Introduction to Fourier Series omposing Functions into Trigonometric Components
	Your Name Department of Mathematics July 9, 2025
Your Name (Departme	et of Mathematica) Introduction to Fearly Servers July 2-005 1
Outline	
Introduce	ction and Motivation
2 Mathem	natical Foundation
3 Example	a: Square Wave
Applicat	tions
Your Name (Departme What are	e Fourier Series?
Main Idea Any period functions w	a ic function can be expressed as a sum of sine and cosine vith different frequencies.
Mad	
For a perio	atical Expression discrete for the second s
For a perio	Atical Expression dic function $f(x)$ with period 2π : $f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos(nx) + b_n \sin(nx))$
For a perio	stical Expression dic function $f(x)$ with period 2π : $f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos(nx) + b_n \sin(nx))$
Your Name Departme	Stical Expression dic function $f(x)$ with period 2π : $f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos(nx) + b_n \sin(nx))$
Wathema For a perio	$\frac{\operatorname{stical Expression}}{f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos(nx) + b_n \sin(nx))$
Valhema For a perio Ver Name (Departme What are Main Idea Any periodi functions w	Stical Expression dic function $f(x)$ with period 2π : $f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos(nx) + b_n \sin(nx))$ et al Mathematica) Expression to Fourier Series c function can be expressed as a sum of sine and cosine ith different frequencies.
Vor Vare (Opener Vor Vare (Opener What are Main Idea Any period functions w Mathema For a perio	stical Expression dic function $f(x)$ with period 2π : $f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos(nx) + b_n \sin(nx))$ (a) Construction (nx) (nx) (nx) (nx) (nx) (nx) (nx) (nx) (nx) (nx) (nx) (nx) (nx
Wathema For a perio Warking (Opport What are Main Idea Any periodi functions w Mathema For a period	stical Expression dic function $f(x)$ with period 2π : $f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos(nx) + b_n \sin(nx))$ end of Mathematical respective to Fourier Series? c function can be expressed as a sum of sine and cosine ith different frequencies. tical Expression dic function $f(x)$ with period 2π : $f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos(nx) + b_n \sin(nx))$

Yo

Name (Department of M

Fourier Coefficients	
The coefficients are determined by:	
$a_0 = \frac{1}{\pi} \int^{\pi} f(x) dx$	(1)
$a_n = \frac{1}{2} \int_{-\pi}^{\pi} f(x) \cos(nx) dx$	(2)
$b_n = \frac{1}{2} \int_{-\pi}^{\pi} f(x) \sin(nx) dx$	(3)
$\pi J_{-\pi}$ Orthogonality	
These formulas arise from the orthogonality of trigonometric functions:	
$\int_{-\infty}^{\pi} \cos(mx) \cos(nx) dx = \begin{cases} 0 & \text{if } m \neq n \end{cases}$	4/8
Square Wave Function	470
Define the square wave: $f(x) = \begin{cases} 1 & \text{if } 0 < x < \pi \\ 1 & \text{if } 0 < x < \pi \end{cases}$	
$f(x) = \begin{cases} -1 & \text{if } -\pi < x < 0 \end{cases}$	
Computing the coefficients: $-3 - 2 - 1 \qquad 1 \qquad 2$	3
$a_0 = 0$ (4) $a_n = 0$ (5)	
$b_n = \frac{4}{n\pi} \pmod{n} \tag{6}$	
Your Name (Department of Mathematics) Introduction to Fourier Series July 9, 2025	8 - O A CH
	0/0
Square Wave Fourier Series	370
Square Wave Fourier Series The Fourier series representation:	3/0
Square Wave Fourier Series The Fourier series representation: $f(x) = \frac{4}{\pi} \sum_{n=1.3.5,}^{\infty} \frac{\sin(nx)}{n}$	979
Square Wave Fourier Series The Fourier series representation: $f(x) = \frac{4}{\pi} \sum_{n=1:3,5,}^{\infty} \frac{\sin(nx)}{n}$ Partial Sums	5,6
Square Wave Fourier Series The Fourier series representation: $f(x) = \frac{4}{\pi} \sum_{n=1,3,5,}^{\infty} \frac{\sin(nx)}{n}$ Partial Sums $S_1(x) = \frac{4}{\pi} \sin(x)$	(7)
Square Wave Fourier Series The Fourier series representation: $f(x) = \frac{4}{\pi} \sum_{n=1:3,5,}^{\infty} \frac{\sin(nx)}{n}$ Partial Sums $S_1(x) = \frac{4}{\pi} \sin(x)$ $S_3(x) = \frac{4}{\pi} \left(\sin(x) + \frac{\sin(3x)}{2} \right)$	(7)
Square Wave Fourier Series The Fourier series representation: $f(x) = \frac{4}{\pi} \sum_{n=1,3,5,}^{\infty} \frac{\sin(nx)}{n}$ Partial Sums $S_1(x) = \frac{4}{\pi} \sin(x)$ $S_3(x) = \frac{4}{\pi} \left(\sin(x) + \frac{\sin(3x)}{3}\right)$ $S_5(x) = \frac{4}{\pi} \left(\sin(x) + \frac{\sin(3x)}{3} + \frac{\sin(5x)}{3}\right)$	(7) (8) (9)
Square Wave Fourier Series The Fourier series representation: $f(x) = \frac{4}{\pi} \sum_{n=1,3,5,}^{\infty} \frac{\sin(nx)}{n}$ Partial Sums $S_1(x) = \frac{4}{\pi} \sin(x)$ $S_3(x) = \frac{4}{\pi} \left(\sin(x) + \frac{\sin(3x)}{3}\right)$ $S_5(x) = \frac{4}{\pi} \left(\sin(x) + \frac{\sin(3x)}{3} + \frac{\sin(5x)}{5}\right)$	(7) (8) (9)
Square Wave Fourier Series The Fourier series representation: $f(x) = \frac{4}{\pi} \sum_{n=1,3,5,}^{\infty} \frac{\sin(nx)}{n}$ Partial Sums $S_1(x) = \frac{4}{\pi} \sin(x)$ $S_3(x) = \frac{4}{\pi} \left(\sin(x) + \frac{\sin(3x)}{3}\right)$ $S_5(x) = \frac{4}{\pi} \left(\sin(x) + \frac{\sin(3x)}{3} + \frac{\sin(5x)}{5}\right)$ Chevergence	(7) (8) (9)
Square Wave Fourier Series The Fourier series representation: $f(x) = \frac{4}{\pi} \sum_{n=1,3,5,}^{\infty} \frac{\sin(nx)}{n}$ Partial Sums $S_1(x) = \frac{4}{\pi} \sin(x)$ $S_3(x) = \frac{4}{\pi} \left(\sin(x) + \frac{\sin(3x)}{3}\right)$ $S_5(x) = \frac{4}{\pi} \left(\sin(x) + \frac{\sin(3x)}{3} + \frac{\sin(5x)}{5}\right)$ Convergence We have (represented reference) in the convergence in	(7) (8) (9)
Square Wave Fourier Series The Fourier series representation: $f(x) = \frac{4}{\pi} \sum_{n=1,3,5,}^{\infty} \frac{\sin(nx)}{n}$ Partial Sums $S_1(x) = \frac{4}{\pi} (\sin(x) + \frac{\sin(3x)}{3})$ $S_3(x) = \frac{4}{\pi} (\sin(x) + \frac{\sin(3x)}{3})$ $S_5(x) = \frac{4}{\pi} (\sin(x) + \frac{\sin(3x)}{3} + \frac{\sin(5x)}{5})$ Moreover Convergence Two regions of Fourier Series Signal Processing = Audio compression (MP3) = Image processing (JPEG) = Digital communications Digital communications Heat conduction = Vibration analysis = Electrical circuits = Solving differential equations = Solving differential equations	(7) (8) (9)

luly 9, 2025

7/8

Figure 14.2: Slides showing the square wave function and its Fourier series approximation.

_Case Study: From Conjecture to

Conclusion

15

Tis much better to do a little with certainty and leave the rest for others that come after than to explain all things by conjecture without making sure of any thing.ugh. — Isaac Newton (1643–1727)

He who knows that enough is enough will always have enough. 知足者常乐。

- Laozi (老子)

IN THE PRECEDING CHAPTERS, we have explored the individual components of an AI-assisted research workflow. We have seen how to use AI for brainstorming, code generation, verification, and document preparation. This chapter aims to synthesize these skills by walking through a complete, albeit miniature, research project from start to finish.

We will follow a realistic trajectory: a vague curiosity is honed into a specific question with AI assistance; this question is investigated computationally; the resulting pattern is visualized and formally stated; and finally, the finding is prepared for dissemination with the appropriate ethical disclosures. This case study will demonstrate the role of the human researcher as the director of the project, using AI as a powerful and versatile tool at each stage.

15.1 The Spark of Curiosity: Problem Formulation

OUR PROJECT BEGINS NOT WITH A CLEAR HYPOTHESIS, but with a general interest in a fascinating area of combinatorics: integer partitions. As a student entering this field, a great first step is to ask an AI to survey the landscape for accessible yet interesting problems. This leverages the LLM's broad training data to act as a guide.

Example 15.1 (Brainstorming a Research Question). We start by prompting an AI to suggest research directions suitable for an undergraduate. This is an application of the techniques discussed in Chapter 9.

Prompt to Al

I'm working on a project about integer partitions. Can you suggest some interesting, accessible research questions or potential conjectures for an undergraduate to investigate? For example, related to partitions with specific restrictions. Brainstorm to find a question

Al Output

Certainly! Here are a few accessible directions in the theory of integer partitions:

- 1. **Congruence Properties:** Ramanujan discovered that $p(5k + 4) \equiv 0 \pmod{5}$. Are there similar, undiscovered congruence properties for partitions with specific restrictions? For example, let a(n) be the number of partitions of n into parts that are all odd and distinct. Does the sequence $a(n) \pmod{3}$ have any interesting properties?
- 2. **Bridging Different Restrictions:** Explore potential connections between two different types of restricted partitions. This relates to Rogers-Ramanujan type identities.
- 3. **Graphical Representation:** Investigate the properties of Ferrers diagrams for certain classes of partitions, such as the size of the Durfee square.

The first suggestion is particularly intriguing. It gives us a concrete function, a(n), and a specific modulus to investigate. This transforms our vague interest into a testable hypothesis: *Does the sequence* $a(n) \pmod{3}$ *exhibit a pattern?*

15.2 Computational Exploration: Code Generation

WITH A CLEAR QUESTION IN HAND, we move from conceptualization to experimentation. We need to generate data to see if a pattern exists. This is a perfect task for AI-assisted code generation, as discussed in Chapter 15.2. We need a script to compute the first several terms of the sequence a(n) (mod 3).

Example 15.2 (Generating an Experimental Script). We ask the AI to write a Python script for our experiment.

Prompt to Al

Write a Python script to compute a(n), the number of partitions of n into odd, distinct parts, for n from 1 to 60. Then, print the sequence of $a(n) \pmod{3}$.

Al Output

```
5 import sympy
```

Using a dictionary for memoization to speed up computation memo = {} Generate an Experimental Script

```
def a(n):
    """Computes the number of partitions of n into odd,
        distinct parts."""
    if n in memo:
       return memo[n]
    if n == 0:
        return 1
   if n < 0:
       return 0
    # Using generating functions: product_{k=1 to inf} (1 + q
        ^(2k-1))
    # A direct dynamic programming approach is clearer here.
    parts = [i for i in range(1, n + 1) if i % 2 != 0]
    dp = [0] * (n + 1)
    dp[0] = 1
    for part in parts:
        for i in range(n, part - 1, -1):
            dp[i] += dp[i - part]
    memo[n] = dp[n]
    return dp[n]
if ___name___ == '___main___':
    print("Sequence of a(n) mod 3 for n=1 to 60:")
    results = [str(a(i) \% 3) for i in range(1, 61)]
    # Print in blocks of 8 for easy pattern recognition
    for i in range(0, len(results), 8):
        print(" ".join(results[i:i+8]))
```

Running this code produces the following output:

Output indicating periodicity.

A clear pattern emerges! The sequence $a(n) \pmod{3}$ appears to be periodic with a period of 8.

15.3 VERIFICATION AND VISUALIZATION

THE COMPUTATIONAL EXPERIMENT PROVIDES STRONG EVIDENCE, but it is not a proof. The next steps are to validate this pattern and visualize it for clear communication, drawing on the skills from Chapters 15.3 and 12.

First, we can ask the AI to help us find a path towards a formal proof. This moves beyond simple code generation into a query about mathematical reasoning.

Prompt to AI

The sequence $a(n) \pmod{3}$ appears to be periodic with period 8. Can you provide a proof sketch or a key idea for how to prove this, perhaps using generating functions?

Al Output

This is an excellent observation. The standard way to prove such congruence properties is through the generating function for a(n), which is:

$$A(q) = \sum_{n=0}^{\infty} a(n)q^n = \prod_{k=1}^{\infty} (1+q^{2k-1})$$

To analyze this modulo 3, we can use the fact that $(1 + x)^3 \equiv 1 + x^3 \pmod{3}$. This suggests we should look at $A(q)^3$. While a full proof is complex and relies on identities like the Jacobi Triple Product, the core idea is to show that the generating function satisfies a specific polynomial identity modulo 3 that implies the periodicity of its coefficients. Investigating the properties of A(q) modulo 3 is the correct path.

This response gives us confidence that the pattern is not a numerical artifact and points us toward established proof techniques. Next, to communicate this finding effectively, we create a visualization.

Example 15.3 (Visualizing the Periodic Pattern). Now you want to visualize the result.

Visualization

Prompt to AI

Using the Python code from before, add Matplotlib code to create a stem plot visualizing the periodicity of $a(n) \pmod{3}$ for n = 1 to 32. Save the plot as a vector graphic.

Al Output

```
import matplotlib.pyplot as plt
```

```
Br # (Assume the function a(n) from the previous example is
defined)
```

Finding a proof.

```
n_vals = range(1, 33)
  mod_vals = [a(n) % 3 for n in n_vals]
  plt.figure(figsize=(12, 6))
  (markers, stemlines, baseline) = plt.stem(n_vals, mod_vals)
  plt.setp(markers, color='red', markersize=8)
  plt.setp(stemlines, color='blue', linewidth=2)
  plt.setp(baseline, color='gray', linewidth=2, linestyle='-')
plt.title(r'Periodicity of $a(n) \pmod{3}$', fontsize=16)
plt.xlabel('$n$', fontsize=12)
p4 plt.ylabel(r'$a(n) \pmod{3}$', fontsize=12)
  plt.xticks(range(0, 33, 4))
  plt.yticks([0, 1, 2])
  plt.grid(True, axis='y', linestyle='--')
  # Save as a vector graphic for publication quality
  plt.savefig("partition_periodicity.pdf", format="pdf",
       bbox_inches="tight")
  plt.show()
```

15.4 Dissemination: LATEX and Ethical Disclosure

WE NOW HAVE A COMPUTATIONALLY-SUPPORTED CONJECTURE, a plausible path to a proof, and a clear visualization. The final step is to formally write up our finding, as discussed in Chapters 13 and 16.

First, we use AI to help us typeset our conjecture in formal mathematical language.

Prompt to AI

Generate the LAT_EX code for a 'conjecture' environment stating that for $n \ge 1$, the sequence $a(n) \pmod{3}$ is periodic with period 8, and list the repeating values.

Al Output

23

```
21 \begin{conjecture}
22 Let $a(n)$ be the number of partitions of an integer $n$ into
odd, distinct parts. For $n \ge 1$, the sequence of
recidues of $a(n)$ medule 2 is periodic with period $
20 medule 2 medule 2 is periodic with period $
20 medule 2 m
```

- residues of \$a(n)\$ modulo 3 is periodic with period 8.
 Specifically, for \$k \ge 0\$ and \$1 \le r \le 8\$:
 \[a(8k+r) \equiv a(r) \pmod 3 \]
- The repeating sequence of residues \$\{a(r) \pmod 3\}_{r=1}^8\$
 is \$\{1, 1, 1, 2, 1, 1, 2, 0\}\$.

Generate I₄T_EX code

Finally, and most importantly, we must include a transparent disclosure of the AI's role in our research. This statement would appear in the acknowledgments or methodology section of our paper.

Remark (Example Disclosure Statement). The initial conjecture regarding the periodicity of $a(n) \pmod{3}$ was formulated after an exploratory interaction with Anthropic's Claude 3 Opus, which suggested investigating the sequence as a potentially fruitful research direction. The Python scripts used for numerical verification and visualization were initially generated with assistance from the same AI model and subsequently reviewed, validated, and modified by the author. The AI's suggestion for a proof strategy involving generating functions was used as a starting point for the formal proof.

This case study demonstrates the power of a human-AI partnership. The human researcher directed the inquiry, verified the results, and retained intellectual ownership, while the AI served as a powerful assistant to accelerate discovery, experimentation, and communication. **Disclosure Statement**

16 *Ethics and AI-Assisted Mathematics*

Scientific knowledge is an enabling power to do either good or bad — but it does not carry instructions on how to use it.

— Richard Feynman (1918 – 1988)

The superior man thinks of virtue; the small man thinks of comfort. 君子喻于义,小人喻于利。

— Confucius (孔子)

THE INTEGRATION OF ARTIFICIAL INTELLIGENCE into mathematical research raises profound ethical questions that extend beyond simple concerns about accuracy and reliability. As AI systems become increasingly sophisticated and widely adopted in mathematical practice, the research community must grapple with fundamental questions about intellectual integrity, attribution, transparency, and the changing nature of mathematical knowledge production. These ethical considerations are not merely academic concerns but have practical implications for how mathematical research is conducted, evaluated, and communicated.

The question of intellectual contribution represents one of the most fundamental ethical challenges in AI-assisted mathematics. When an AI system generates a proof, suggests a novel approach, or creates substantial mathematical content, determining appropriate attribution becomes complex. Traditional notions of authorship assume human agency and creativity, but AI systems can produce genuinely useful and even creative mathematical contributions.

Consider the scenario where an AI system suggests a key lemma that enables the completion of a proof. The human researcher may have posed the problem, guided the AI's exploration, and verified the final result, but the crucial insight came from the AI. In such cases, the ethical question becomes: what constitutes sufficient human contribution to warrant sole authorship? The mathematical community has not yet developed clear consensus on these attribution questions, but several principles are emerging as important considerations.

Transparency emerges as a critical ethical imperative in AI-assisted mathematical research. When AI tools contribute to mathematical work, researchers have an ethical obligation to disclose this assistance in their publications. This transparency serves multiple purposes: it allows readers to properly evaluate the work, enables reproducibility, and maintains the integrity of the mathematical record. However, the level of detail required in such disclosures remains a matter of ongoing debate.

16.1 INTELLECTUAL INTEGRITY AND ATTRIBUTION

THE CONCEPT OF INTELLECTUAL INTEGRITY IN MATHEMATICS has traditionally been built around the assumption of human authorship. Mathematical theorems, proofs, and insights were understood to be products of human reasoning, creativity, and effort. The introduction of AI systems that can generate mathematical content challenges this foundational assumption and requires a reconceptualization of what constitutes intellectual contribution.

When evaluating the role of AI in mathematical work, researchers must consider the spectrum of AI assistance. At one end, AI might serve as a sophisticated calculator or computational tool, performing routine calculations or checking arithmetic. At the other end, AI might generate novel mathematical insights, suggest creative proof strategies, or identify unexpected connections between different areas of mathematics. The ethical implications vary significantly across this spectrum.

For routine computational assistance, the ethical requirements are relatively straightforward. Just as mathematicians have always acknowledged the use of computational tools, the use of AI for basic calculations or symbolic manipulation should be noted but does not necessarily require detailed attribution. However, when AI systems contribute substantive mathematical ideas, the ethical obligations become more complex.

Attribution Guidelines for AI Assistance

- **Computational Role**: When AI performs routine calculations, acknowledge in methods section
- Exploratory Role: When AI suggests directions for investigation, note in acknowledgments
- **Creative Role**: When AI generates key insights or proof strategies, consider co-authorship or detailed attribution
- Generative Role: When AI produces substantial mathematical content, full disclosure and potentially shared credit required

The challenge of attribution is further complicated by the collaborative nature of AI-human interaction. Mathematical research assisted by AI is rarely a simple matter of the AI producing output and the human accepting it. Instead, it typically involves iterative dialogue, where the human guides the AI's exploration, refines its suggestions, and integrates its contributions into a broader mathematical framework. This collaborative process makes it difficult to cleanly separate human and AI contributions.

The cardinal rule is simple: **You are the author. You are responsible for the final work.** An AI cannot be a co-author because it cannot be held accountable. Presenting AI-generated text, proofs, or code as your own Attribution Guidelines for AI Assistance without significant intellectual contribution, verification, and modification is plagiarism.

16.2 TRANSPARENCY AND DISCLOSURE REQUIREMENTS

TRANSPARENCY IN AI-ASSISTED MATHEMATICAL RESEARCH serves multiple stakeholders and purposes. For the broader mathematical community, transparency enables proper evaluation of research contributions and maintains the integrity of mathematical knowledge. For reviewers and editors, detailed disclosure allows for appropriate assessment of the work's originality and significance. For future researchers, transparency enables reproducibility and builds upon previous work effectively.

The scope of required disclosure extends beyond simply mentioning that AI tools were used. Ethical transparency requires providing sufficient detail about the nature and extent of AI assistance to allow readers to make informed judgments about the work. This includes information about which AI systems were used, how they were employed, what types of contributions they made, and how their outputs were verified and validated.

Current best practices in transparency are still evolving, but several principles are gaining acceptance. First, the disclosure should be proportional to the AI's contribution. Minor computational assistance might warrant only brief mention, while substantial AI-generated content requires detailed explanation. Second, the disclosure should be specific rather than generic. Instead of simply stating that "AI tools were used," researchers should explain precisely how and where AI assistance was employed.

Common Transparency Failures

- · Vague statements like "AI was used for some calculations"
- Failure to disclose AI assistance in key insights or proof strategies
- Inadequate information about AI tool versions and configurations
- Missing details about verification and validation procedures
- Unclear boundaries between human and AI contributions

The timing and placement of disclosure also raise important considerations. Some journals now require disclosure of AI assistance during the submission process, while others expect it to be included in the published article. The mathematical community is moving toward requiring disclosure both during review and in the final publication, ensuring that AI assistance is transparent throughout the research process.

If an AI played a significant role in your work, you must disclose it.

Common Transparency Failures

This is a matter of academic honesty and reproducibility. The standard is to include a statement in your methods or acknowledgements and a citation in your bibliography.

Example 16.1 (Example In-text Description). "The Python code for the numerical simulation (see Appendix A) was initially generated using OpenAI's GPT-4 model ¹ and subsequently verified and modified by the author."

16.3 VERIFICATION AND CRITICAL EVALUATION

THE ETHICAL USE OF AI IN MATHEMATICAL RESEARCH requires rigorous verification and critical evaluation of AI-generated content. This responsibility cannot be delegated to the AI system itself, regardless of how sophisticated or reliable it may appear. Human mathematicians must maintain ultimate responsibility for the correctness and validity of their research, even when AI systems contribute substantially to the work.

Verification strategies for AI-assisted mathematical research must be more comprehensive than traditional mathematical verification. While traditional mathematical work primarily requires checking logical consistency and computational accuracy, AI-assisted work also requires evaluating the appropriateness of AI contributions, the reliability of AI-generated insights, and the completeness of the mathematical reasoning.

The verification process should begin with a critical assessment of the AI system's capabilities and limitations. Different AI tools have different strengths and weaknesses, and understanding these characteristics is essential for appropriate use. Large language models, for example, may excel at generating plausible-sounding mathematical explanations but may lack the rigorous logical reasoning required for valid proofs. Computer algebra systems may perform symbolic manipulations accurately but may not recognize when their operations are mathematically meaningful.

Verification Checklist for AI-Assisted Mathematics

- Logical Consistency: Does the AI-generated content follow logical rules and mathematical principles?
- Computational Accuracy: Are calculations and symbolic manipulations correct?
- 3. **Conceptual Soundness**: Do AI suggestions align with established mathematical knowledge?
- 4. **Completeness**: Are there gaps in reasoning that the AI may have overlooked?
- 5. **Originality**: Is the AI output genuinely novel or merely recombining existing knowledge?

¹ OpenAI. ChatGPT, 2024. Model: GPT-4. Accessed on: 2024-05-21 6. **Relevance**: Do AI contributions actually address the mathematical question at hand?

The verification process must also consider the broader context of the mathematical research. AI systems may generate locally correct mathematical statements that are nonetheless inappropriate for the specific research context. For example, an AI might suggest a valid mathematical technique that is unnecessarily complex for the problem at hand, or it might provide a correct but uninformative proof that fails to illuminate the underlying mathematical structure.

LLMs invent things. They create plausible but false theorems, proofs, and citations. This is not a bug, but a feature of their probabilistic nature. Your academic duty is to be professionally skeptical. Verify every fact, check every calculation with a CAS, and look up every citation.

Verification Checklist for AI-Assisted Mathematics

16.4 BIAS AND FAIRNESS IN AI-ASSISTED RESEARCH

THE TRAINING DATA AND ALGORITHMS underlying AI systems can introduce biases that affect mathematical research in subtle but important ways. These biases may manifest in the types of mathematical problems AI systems are better equipped to handle, the approaches they tend to suggest, or the mathematical traditions they draw upon. Understanding and addressing these biases is crucial for maintaining fairness and inclusivity in mathematical research.

Training data bias represents a significant concern in AI-assisted mathematical research. AI systems are typically trained on existing mathematical literature, which may reflect historical biases in mathematical publishing, funding, and recognition. If certain mathematical approaches, cultural perspectives, or research areas are underrepresented in the training data, the AI system may be less capable of supporting research in those areas.

The implications of such bias extend beyond individual research projects to the broader development of mathematical knowledge. If AI systems consistently favor certain types of mathematical approaches or problems, they may inadvertently steer the field toward some areas while neglecting others. This could lead to a narrowing of mathematical research or the perpetuation of existing inequalities in mathematical opportunity and recognition.

Potential Sources of Bias in Mathematical AI

- Historical Bias: Training data reflects past inequalities in mathematical publishing
- Language Bias: AI systems may be better equipped for mathematics in certain languages
- **Cultural Bias**: Different mathematical traditions may be unequally represented
- Institutional Bias: Work from certain institutions or regions may be overrepresented
- **Methodological Bias**: Some proof techniques or approaches may be favored over others

Addressing bias in AI-assisted mathematical research requires both individual and collective action. Individual researchers can work to recognize and compensate for potential biases in their AI tools, seeking diverse perspectives and approaches in their mathematical work. The mathematical community as a whole must work to ensure that AI development includes diverse voices and perspectives, and that the benefits of AI-assisted research are distributed equitably.

AI models are trained on vast datasets from the internet. While less of a concern in pure mathematics than in other fields, they may still reflect biases present in their training data (e.g., in historical narratives or by over-representing certain research directions). Be aware of this potential.

16.5 The Future of Mathematical Authorship

THE INTEGRATION OF AI INTO MATHEMATICAL RESEARCH is fundamentally changing the nature of mathematical authorship and contribution. As AI systems become more sophisticated and capable of generating substantial mathematical content, the traditional model of individual human authorship may need to evolve to accommodate new forms of collaboration between humans and AI systems.

Several models for AI-human collaboration in mathematical authorship are emerging. The traditional model treats AI as a tool, similar to computational software or mathematical instruments. In this model, the human researcher remains the sole author, but acknowledges AI assistance in the methodology or acknowledgments section. This approach works well when AI contributions are primarily computational or supportive.

An alternative model recognizes AI systems as collaborators rather than tools. In this model, AI contributions are acknowledged more prominently, potentially including AI systems in author lists or providing detailed attribution for AI-generated content. This approach becomes more appropriate Potential Sources of Bias in Mathematical AI as AI systems begin to generate substantial mathematical insights and creative contributions.

Future Models of AI-Human Mathematical Collaboration

- AI as Tool: Traditional authorship with AI assistance acknowledged
- AI as Collaborator: Shared credit for AI contributions to mathematical insights
- AI as Co-author: Formal recognition of AI systems in authorship
- **Hybrid Models**: Flexible approaches based on the nature and extent of AI contribution

The evolution of mathematical authorship will likely require new standards and practices for evaluating mathematical contributions. Traditional metrics for assessing mathematical research, such as originality, significance, and rigor, may need to be refined to account for AI assistance. New metrics might consider the quality of human-AI collaboration, the appropriateness of AI tool selection, and the effectiveness of verification and validation procedures.

The line between a tool and a collaborator is becoming blurred. The academic community is still developing standards for what constitutes appropriate use versus co-authorship. For now, the consensus is clear: humans are authors, and AIs are tools to be cited. This may evolve, but your guiding principle should always be intellectual honesty.

16.6 ETHICAL GUIDELINES FOR PRACTICE

BASED ON THE ETHICAL CONSIDERATIONS DISCUSSED ABOVE, several practical guidelines emerge for the responsible use of AI in mathematical research. These guidelines are not absolute rules but rather principles that should be adapted to specific research contexts and evolving technological capabilities.

First, researchers should maintain transparency about AI assistance throughout the research process. This includes documenting AI tool usage, preserving records of AI interactions, and providing clear disclosure in publications. The level of detail should be proportional to the significance of AI contributions, with more substantial AI assistance requiring more detailed disclosure.

Second, researchers must take responsibility for verification and validation of AI-generated content. This responsibility cannot be delegated to AI systems themselves, regardless of their apparent reliability. Human researchers must develop and maintain the skills necessary to critically evaluate AI contributions and ensure the correctness of their mathematical Future Models of AI-Human Mathematical Collaboration work.

Third, researchers should consider the broader implications of their AI use for mathematical knowledge and practice. This includes being mindful of potential biases in AI systems, considering the impact of AI use on mathematical education and training, and working to ensure that AI benefits are distributed equitably across the mathematical community.

Ethical Guidelines Checklist

- 1. Transparency: Clearly document and disclose AI assistance
- 2. Verification: Rigorously validate all AI-generated content
- 3. Attribution: Provide appropriate credit for AI contributions
- 4. **Responsibility**: Maintain human accountability for mathematical correctness
- 5. Fairness: Consider and address potential biases in AI systems
- 6. Education: Maintain and develop human mathematical capabilities
- Community: Consider broader implications for mathematical practice

Power requires responsibility. Using these tools effectively means using them ethically.

The decision tree in Figure 16.1 provides a practical framework for navigating the ethical complexities of AI use in mathematical research. The flowchart distinguishes between different types of AI assistance, from brainstorming and technical help (which are generally appropriate with proper disclosure) to core research claims (which require extreme caution). The key insight is that the appropriateness of AI use depends heavily on the specific application and the researcher's ability to verify the results independently. Notice that the tree includes an alternative path for content generation—rather than using AI to generate mathematical proofs directly, students should use AI to generate testable hypotheses, then conduct the research themselves using reliable tools.

Figure 16.2 provides a practical decision-making model for students.

Ethical Guidelines Checklist


Figure 16.1: Decision tree for ethical AI use in mathematical research. The flowchart guides students through key decision points, from initial tool selection to final documentation. Color coding indicates appropriate use (green), situations requiring careful verification (yellow), and inappropriate use (red). The tree emphasizes that core research insights must originate from human reasoning, while AI can appropriately assist with technical tasks.



Figure 16.2: A decision-making flowchart for the ethical use of AI in research.

Research Methodology and Best

Practices

17

It is not knowledge, but the act of learning, not possession but the act of getting there, which grants the greatest enjoyment.

- Carl Friedrich Gauss (1777 - 1855)

A journey of a thousand miles begins with a single step. 千里之行,始于足下。

— Laozi (老子)

THE INTEGRATION OF AI INTO MATHEMATICAL RESEARCH requires the development of new methodological approaches that effectively combine human expertise with artificial intelligence capabilities. This chapter examines best practices for incorporating AI into mathematical research workflows, from initial problem formulation through publication and dissemination. The goal is not to replace traditional mathematical research methods but to augment them with AI capabilities while maintaining the rigor and integrity that define mathematical scholarship.

Effective AI-assisted mathematical research begins with careful problem formulation and planning. Unlike traditional mathematical research, which often follows a relatively linear progression from problem identification to solution, AI-assisted research requires more iterative and exploratory approaches. Researchers must consider not only the mathematical aspects of their problem but also the capabilities and limitations of available AI tools.

The problem formulation phase should include an assessment of which aspects of the research problem are most suitable for AI assistance. Some mathematical tasks, such as symbolic computation, pattern recognition, or exhaustive case analysis, may be well-suited to AI assistance. Others, such as developing mathematical intuition, making creative leaps, or providing conceptual understanding, may require primarily human involvement.

17.1 Planning AI-Assisted Research Projects

STRATEGIC PLANNING BECOMES CRUCIAL when incorporating AI into mathematical research. Researchers must develop comprehensive plans that account for both the mathematical and technological aspects of their projects. This planning should include clear objectives for AI use, selection of appropriate AI tools, development of verification strategies, and consideration of ethical implications.

The planning process should begin with a thorough analysis of the

mathematical problem and its structure. This analysis should identify specific tasks or subproblems that might benefit from AI assistance, as well as areas where human expertise will be essential. For example, a research project in number theory might benefit from AI assistance in computational searches for examples or counterexamples, while requiring human insight for developing theoretical frameworks and proofs.

Tool selection represents a critical component of research planning. Different AI systems have different strengths and limitations, and matching the right tool to the right task is essential for effective research. Large language models might excel at generating mathematical explanations or suggesting proof strategies, while computer algebra systems are better suited for symbolic computation and algebraic manipulation. Machine learning systems might be appropriate for pattern recognition tasks, while theorem provers are designed for formal verification.

Research Planning Framework

- 1. **Problem Analysis**: Identify mathematical structure and requirements
- 2. **Task Decomposition**: Break problem into AI-suitable and human-suitable components
- 3. Tool Selection: Choose appropriate AI systems for each task
- Verification Strategy: Plan methods for validating AI contributions
- 5. **Integration Approach**: Design workflow for combining human and AI work
- 6. **Ethical Considerations**: Address transparency, attribution, and responsibility

The integration approach requires careful consideration of how human and AI contributions will be combined throughout the research process. This may involve iterative cycles of AI-generated content followed by human analysis and refinement, or it may involve more concurrent collaboration where AI systems provide real-time assistance during human mathematical work.

When planning a project, explicitly define the roles AI will play. Will it be used for literature discovery, code generation, or text refinement? A clear plan prevents over-reliance and ensures the core intellectual work remains your own. Treat the AI as a specific tool for a specific task. Research Planning Framework

17.2 Collaborative Workflows and Documentation

EFFECTIVE AI-ASSISTED MATHEMATICAL RESEARCH requires robust documentation and workflow management. Unlike traditional mathematical research, where the primary documentation consists of mathematical notation and reasoning, AI-assisted research must also document the AI tools used, the nature of AI contributions, and the verification processes employed.

Documentation should begin from the earliest stages of research and continue throughout the project. This includes maintaining records of AI tool configurations, preserving AI-generated content, documenting verification procedures, and tracking the evolution of mathematical ideas through human-AI collaboration. Such documentation serves multiple purposes: it enables reproducibility, supports transparency requirements, and provides a foundation for future research.

The documentation process should capture not only the final results of AI assistance but also the iterative process of human-AI collaboration. This includes recording failed attempts, documenting the reasoning behind AI tool choices, and preserving the context in which AI contributions were made. Such comprehensive documentation provides valuable insights for both the current research project and future work.

Documentation Requirements

- AI Tool Information: Versions, configurations, and capabilities
- Interaction Records: Prompts, responses, and iterative refinements
- Verification Procedures: Methods used to validate AI contributions
- Integration Process: How AI and human contributions were combined
- **Decision Rationale**: Reasoning behind AI tool selection and usage

Collaborative workflows must also account for the different timescales and working patterns of human and AI contributors. While AI systems can generate responses quickly, human evaluation and integration of AI contributions typically requires more time and reflection. Effective workflows balance the speed of AI generation with the deliberation required for rigorous mathematical reasoning.

When working in a team, establish clear guidelines for AI use. All members should document when and how they used AI tools. For reproducibility, it is good practice to save important AI conversations or generated code snippets in a project repository, along with notes on how Documentation Requirements

they were verified and modified.

17.3 QUALITY ASSURANCE AND PEER REVIEW

THE INTEGRATION OF AI INTO MATHEMATICAL RESEARCH necessitates enhanced quality assurance procedures that go beyond traditional mathematical peer review. While traditional peer review focuses primarily on mathematical correctness and significance, AI-assisted research requires additional evaluation of AI tool usage, verification procedures, and the appropriateness of human-AI collaboration.

Quality assurance should begin with self-evaluation by the research team. This includes systematic verification of AI-generated content, assessment of the appropriateness of AI tool usage, and evaluation of the overall integration of human and AI contributions. Researchers should develop internal protocols for reviewing AI-assisted work before submission for external review.

The peer review process for AI-assisted mathematical research may require reviewers with expertise in both mathematical content and AI systems. Traditional mathematical reviewers may not have the technical knowledge necessary to evaluate AI tool usage, while AI experts may lack the mathematical expertise to assess the mathematical content. This suggests the need for expanded review teams or reviewers with interdisciplinary expertise.

Extended Review Criteria for AI-Assisted Mathematics

- Mathematical Correctness: Traditional verification of mathematical content
- AI Tool Appropriateness: Assessment of AI system selection and usage
- Verification Adequacy: Evaluation of validation procedures
- Transparency Compliance: Review of disclosure and documentation
- Integration Quality: Assessment of human-AI collaboration effectiveness
- Ethical Compliance: Review of ethical considerations and guidelines

AI-generated content requires an even more stringent review process. When you receive code or text from an AI, treat it as a submission from a new, un-trusted collaborator. You must perform a thorough peer review of its output before incorporating it into your work. This means checking logic, verifying facts, and testing code rigorously. Extended Review Criteria for AI-Assisted Mathematics

17.4 Publication and Dissemination Strategies

THE PUBLICATION OF AI-ASSISTED MATHEMATICAL RESEARCH requires careful consideration of disclosure requirements, formatting standards, and dissemination strategies. Different journals and conferences are developing varying standards for AI-assisted research, and researchers must navigate these evolving requirements while maintaining transparency and integrity.

Publication strategies should consider the appropriate venues for AIassisted mathematical research. Some journals may be more welcoming of AI-assisted work, while others may require more extensive justification or documentation. Researchers should familiarize themselves with the policies and expectations of their target publications before submitting AI-assisted work.

The presentation of AI-assisted research in publications requires careful attention to clarity and completeness. Readers should be able to understand the nature and extent of AI assistance, the verification procedures employed, and the integration of human and AI contributions. This may require new formatting conventions or supplementary materials that document AI assistance in detail.

When submitting your work for publication, be upfront about your use of AI. Check the journal or conference's specific policies on AI disclosure. A transparent approach builds trust with reviewers and readers and contributes to the community's understanding of these new methodologies.

____Future Directions and Emerging

Technologies

18

The future is not some place we are going to, but one we are creating. The paths are not to be found, but made, and the activity of making them changes both the maker and the destination.

— John Schaar (1928 – 2011)

Mathematics is the language that describes the fundamental structure of the universe. 数学是描述宇宙基本结构的语言。

— Yau Shing-Tung (丘成桐) (1949 -)

THE LANDSCAPE OF AI-ASSISTED MATHEMATICAL RESEARCH CONTIN-UES to evolve rapidly, with new technologies and methodologies emerging regularly. Understanding these developments and their potential implications is crucial for researchers who want to effectively incorporate AI into their work and for the mathematical community as a whole to navigate the changing research environment.

Recent advances in AI technology suggest several promising directions for mathematical research. Large language models are becoming increasingly sophisticated in their ability to understand and generate mathematical content, while specialized AI systems are being developed specifically for mathematical reasoning and proof generation. Computer algebra systems are incorporating more AI capabilities, and new hybrid systems are emerging that combine multiple AI approaches.

The development of multimodal AI systems that can process mathematical notation, diagrams, and natural language simultaneously represents a significant advancement for mathematical research. These systems can potentially understand mathematical content in a more holistic way, similar to how human mathematicians integrate visual, symbolic, and conceptual information.

18.1 Emerging AI Technologies for Mathematics

SEVERAL CATEGORIES OF EMERGING AI TECHNOLOGIES show particular promise for mathematical research. Advanced theorem provers are being developed that can generate and verify mathematical proofs with increasing sophistication. These systems go beyond traditional automated reasoning to incorporate heuristic search strategies, machine learning techniques, and large-scale mathematical knowledge bases.

Neural theorem proving represents a particularly exciting development, combining the logical rigor of formal proof systems with the pattern recognition capabilities of neural networks. These systems can potentially discover new proof strategies, identify relevant lemmas, and generate mathematical insights that would be difficult for traditional automated reasoning systems.

Emerging AI Technologies for Mathematics

- Neural Theorem Provers: AI systems that combine logical reasoning with machine learning
- Multimodal Mathematical AI: Systems that process text, symbols, and diagrams together
- Automated Conjecture Generation: AI that proposes new mathematical hypotheses
- Interactive Proof Assistants: AI-enhanced tools for collaborative proof development
- Mathematical Discovery Systems: AI that identifies patterns and generates insights

Automated conjecture generation represents another promising area of development. AI systems are being developed that can analyze mathematical structures, identify patterns, and propose new conjectures for human investigation. These systems have the potential to accelerate mathematical discovery by suggesting new directions for research and highlighting previously unnoticed connections between different areas of mathematics.

The field is moving incredibly fast. We are seeing the development of AI models specifically trained on mathematical and scientific literature. Furthermore, the combination of LLMs with formal proof assistants (like Lean) is a major area of research, with the goal of creating AIs that can help discover and verify novel proofs. Landmark results like DeepMind's AlphaTensor ¹ show that AI can already discover new and more efficient algorithms.

18.2 Integration with Traditional Mathematical Tools

THE FUTURE OF AI-ASSISTED MATHEMATICAL RESEARCH lies not in replacing traditional mathematical tools but in integrating AI capabilities with established mathematical software and methodologies. This integration is happening at multiple levels, from AI-enhanced computer algebra systems to intelligent mathematical editors that provide real-time assistance during mathematical writing.

Modern computer algebra systems are incorporating AI capabilities that enhance their traditional symbolic computation abilities. These enhanced systems can provide more intuitive interfaces, suggest appropriate Emerging AI Technologies for Mathematics

¹ Alhussein Fawzi, Matej Balog, et al. Discovering faster matrix multiplication algorithms with reinforcement learning, 2022 computational strategies, and offer explanations for their operations. The integration of AI with traditional mathematical software creates more powerful and accessible tools for mathematical research.

The development of AI-enhanced mathematical editors and document preparation systems represents another important area of integration. These tools can provide real-time feedback on mathematical writing, suggest improvements to mathematical exposition, and help maintain consistency in mathematical notation and terminology throughout large documents.

The future is not about replacing tools like Mathematica or MATLAB, but integrating them. Expect to see more AI environments that can seamlessly call upon a CAS for reliable calculations, write code for a numerical environment like MATLAB, and then summarize the results in natural language, combining the strengths of all platforms.

18.3 CHALLENGES AND LIMITATIONS

DESPITE THE PROMISING DEVELOPMENTS IN AI-ASSISTED MATHEMAT-ICAL RESEARCH, significant challenges and limitations remain. Understanding these limitations is crucial for the responsible and effective use of AI in mathematical research.

One of the primary challenges is the difficulty of ensuring mathematical rigor in AI-generated content. While AI systems can generate mathematically plausible content, they may lack the deep understanding of mathematical logic and structure necessary for rigorous mathematical reasoning. This challenge requires ongoing development of verification and validation techniques.

The scalability of AI-assisted mathematical research presents another significant challenge. While AI systems may work well for specific mathematical tasks or domains, extending their capabilities to broader areas of mathematics or more complex reasoning tasks remains difficult. This limitation affects the generalizability of AI-assisted research techniques.

Current Limitations of Mathematical AI

- Rigor Verification: Difficulty ensuring mathematical correctness
- **Scalability Issues**: Limited ability to handle complex mathematical reasoning
- Context Understanding: Challenges in maintaining mathematical context across long proofs
- **Creative Insight**: Limited capacity for genuine mathematical creativity
- Interdisciplinary Integration: Difficulty connecting mathematics with other fields

The core challenge remains reasoning. While AIs are becoming better at multi-step logical tasks, true abstract reasoning and the creative leap of insight that defines much of mathematical discovery remain human domains. The "hallucination" problem will likely persist, requiring constant human oversight.

18.4 Implications for Mathematical Education

THE INTEGRATION OF AI INTO MATHEMATICAL RESEARCH has significant implications for mathematical education at all levels. Students entering mathematics research must develop new skills for working with AI systems while maintaining traditional mathematical competencies. This creates challenges for curriculum development and pedagogical approaches.

Mathematical education must balance the benefits of AI assistance with the need for students to develop fundamental mathematical skills and understanding. There is a risk that over-reliance on AI tools could lead to a reduction in students' mathematical problem-solving abilities or conceptual understanding.

The development of AI literacy among mathematics students becomes increasingly important. Students need to understand the capabilities and limitations of AI systems, learn to use AI tools effectively, and develop the judgment necessary to evaluate AI-generated content critically.

Education will need to adapt, shifting focus from rote calculation (which can be automated) to conceptual understanding, critical thinking, and the skill of asking good questions. Curricula may soon include "AI literacy" as a core competency for STEM students. *Current Limitations of Mathematical AI*

18.5 LONG-TERM VISION FOR AI-ASSISTED MATHEMATICS

LOOKING TOWARD THE FUTURE, SEVERAL LONG-TERM POSSIBILITIES emerge for AI-assisted mathematical research. The development of truly collaborative AI systems that can engage in extended mathematical dialogue with human researchers could fundamentally change the nature of mathematical research. Such systems might serve as intelligent research partners, contributing insights and suggestions throughout the research process.

The possibility of AI systems that can generate genuinely novel mathematical insights raises profound questions about the nature of mathematical discovery and creativity. If AI systems can discover new mathematical truths or develop innovative proof techniques, the relationship between human mathematical intuition and mechanical reasoning may need to be reconsidered.

The integration of AI with other emerging technologies, such as quantum computing or advanced visualization systems, could create new possibilities for mathematical research that are difficult to imagine today. These technological convergences may enable investigations into mathematical problems that are currently intractable.

The long-term vision is a human-AI partnership where the mathematician acts as the strategist, posing questions, guiding the exploration, and verifying key insights. The AI will serve as a tireless tactical assistant, exploring vast possibility spaces, handling complex computations, and managing data, freeing up human researchers to operate at a higher level of abstraction.

Conclusion and Recommendations

The important thing is not to stop questioning. Curiosity has its own reason for existing.

— Albert Einstein (1879 – 1955)

The best time to plant a tree was 20 years ago. The second best time is now. 种树最好的时间是二十年前,其次是现在。

— Chinese Proverb (中国谚语)

THE INTEGRATION OF ARTIFICIAL INTELLIGENCE INTO MATHEMATICAL RESEARCH represents a transformative development that offers significant opportunities while presenting important challenges. Throughout this guide, we have explored various aspects of AI-assisted mathematical research, from practical techniques for using AI tools to ethical considerations and future directions. As we conclude, it is important to synthesize these insights into actionable recommendations for senior undergraduate mathematics students beginning their research careers.

The most fundamental recommendation is to approach AI-assisted mathematical research with both enthusiasm and critical judgment. AI tools offer powerful capabilities that can enhance mathematical research in numerous ways, from computational assistance to insight generation. However, these tools must be used thoughtfully, with careful attention to their limitations and appropriate verification of their outputs.

Developing competency in AI-assisted mathematical research requires building skills in multiple areas. Students must maintain and develop their traditional mathematical abilities while also learning to work effectively with AI systems. This includes understanding the capabilities and limitations of different AI tools, developing effective prompting strategies, and building robust verification and validation procedures.

19.1 Key Principles for Responsible AI Use

SEVERAL KEY PRINCIPLES SHOULD GUIDE THE RESPONSIBLE USE of AI in mathematical research. First, transparency must be maintained throughout the research process. This includes documenting AI tool usage, preserving records of AI interactions, and providing clear disclosure in publications. Transparency serves multiple purposes: it enables reproducibility, supports peer review, and maintains the integrity of mathematical scholarship.

Second, human responsibility for mathematical correctness cannot be delegated to AI systems. Regardless of how sophisticated AI tools become, human mathematicians must maintain ultimate responsibility for the validity and correctness of their research. This requires developing and maintaining the skills necessary to critically evaluate AI-generated content.



Third, ethical considerations must be integrated into all aspects of AI-assisted research. This includes considering issues of attribution, addressing potential biases in AI systems, and being mindful of the broader implications of AI use for mathematical knowledge and practice.

Fundamental Principles for AI-Assisted Mathematics

- 1. **Maintain Transparency**: Document and disclose AI assistance appropriately
- 2. Ensure Human Responsibility: Take accountability for mathematical correctness
- 3. **Practice Ethical Use**: Consider attribution, bias, and community impact
- 4. Verify Rigorously: Validate all AI-generated content thoroughly
- 5. **Integrate Thoughtfully**: Combine AI capabilities with human expertise effectively
- 6. **Continue Learning**: Stay current with evolving AI technologies and best practices
- 1. You are the authority. The final responsibility for correctness and integrity is yours.
- 2. Verify, then trust. Never accept an AI's output without independent verification.
- 3. Use AI to augment, not replace, your thinking. The goal is to enhance your skills, not to atrophy them.
- 4. Be transparent. Disclose your use of AI tools in your work.

19.2 BUILDING AI LITERACY FOR MATHEMATICAL RESEARCH

SUCCESS IN AI-ASSISTED MATHEMATICAL RESEARCH requires developing comprehensive AI literacy that encompasses both technical understanding and practical skills. This literacy includes understanding the underlying technologies that power AI systems, recognizing their capabilities and limitations, and developing the judgment necessary to use them effectively.

Technical AI literacy for mathematics students should include understanding the basic principles of machine learning, natural language processing, and automated reasoning. Students need not become AI experts, but they should understand enough about these technologies to use them effectively and evaluate their outputs critically.

Practical AI literacy involves developing skills in prompting AI systems, interpreting AI outputs, and integrating AI assistance into mathemati-

Fundamental Principles for AI-Assisted Mathematics cal workflows. These skills require practice and experience, and students should seek opportunities to work with AI tools in low-stakes environments before applying them to significant research projects.

To be an effective modern mathematician, you must become literate in these tools. This means actively experimenting with different platforms, learning the art of prompt engineering, and understanding the fundamental limitations of the technology. This guide is a starting point for that journey.

The most successful researchers will be those who develop intelligent workflows that integrate AI. This means knowing when to use an LLM for a creative task, when to use a CAS for a rigorous calculation, and when to rely on traditional methods. The LLM chaining technique is a prime example of such a workflow.

19.3 Developing Effective Research Workflows

EFFECTIVE AI-ASSISTED MATHEMATICAL RESEARCH requires developing systematic workflows that combine human expertise with AI capabilities. These workflows should be designed to maximize the benefits of AI assistance while maintaining mathematical rigor and integrity.

The development of effective workflows begins with careful planning and problem analysis. Researchers should identify which aspects of their research problems are most suitable for AI assistance and select appropriate tools for each task. This requires understanding both the mathematical structure of the problem and the capabilities of available AI systems.

Iterative refinement represents a key component of effective AI-assisted research workflows. Rather than viewing AI assistance as a one-time input, researchers should engage in ongoing dialogue with AI systems, using their outputs as starting points for further investigation and refinement.

The tools you use today will be outdated in a few years. The most important skill you can develop is not mastery of a single platform, but the adaptability to learn and integrate new technologies as they emerge. Stay curious and engaged with the evolution of the field.

19.4 Preparing for Future Developments

THE FIELD OF AI-ASSISTED MATHEMATICAL RESEARCH continues to evolve rapidly, and students entering the field must be prepared to adapt to new technologies and methodologies. This requires maintaining a learning mindset and staying current with developments in both mathematical research and AI technology.

Preparation for future developments includes building strong foundational skills in both mathematics and AI that can adapt to new tools and techniques. Rather than focusing exclusively on current AI systems, students should develop the underlying knowledge and skills that will enable them to work effectively with future AI technologies.

Professional development in AI-assisted mathematical research should include engagement with the broader community of researchers working in this area. This includes attending conferences, participating in workshops, and contributing to the ongoing development of best practices and ethical guidelines.

19.5 Contributing to the Mathematical Community

STUDENTS ENTERING AI-ASSISTED MATHEMATICAL RESEARCH have an opportunity to contribute to the development of this emerging field. This includes participating in the establishment of best practices, contributing to the development of ethical guidelines, and helping to shape the future direction of AI-assisted mathematical research.

Contributing to the mathematical community also involves sharing knowledge and experiences with other researchers. This includes publishing research that demonstrates effective use of AI tools, sharing methodological innovations, and contributing to the broader conversation about the role of AI in mathematical research.

The mathematical community benefits from diverse perspectives and approaches to AI-assisted research. Students from different mathematical backgrounds and with different research interests can contribute unique insights and help ensure that AI benefits are distributed broadly across the mathematical community.

As you become proficient, share your knowledge. Discuss best practices with your peers, help develop ethical guidelines in your department, and contribute to the community's collective understanding of how to best leverage these powerful new capabilities for the advancement of mathematics.

19.6 FINAL THOUGHTS

THE INTEGRATION OF AI INTO MATHEMATICAL RESEARCH represents both an opportunity and a responsibility. The opportunity lies in the potential for AI to enhance mathematical research, accelerate discovery, and make mathematical tools more accessible. The responsibility lies in ensuring that these benefits are realized while maintaining the integrity, rigor, and ethical standards that define mathematical scholarship.

Success in AI-assisted mathematical research requires more than technical proficiency with AI tools. It requires developing the judgment to use these tools appropriately, the skills to verify and validate their outputs, and the ethical sensibility to consider their broader implications. Most importantly, it requires maintaining the curiosity, creativity, and dedication to understanding that drive mathematical research.

As AI technologies continue to evolve, the specific tools and techniques discussed in this guide will undoubtedly change. However, the fundamental principles of transparency, responsibility, and ethical use will remain constant. Students who internalize these principles while developing practical skills with current AI tools will be well-positioned to contribute meaningfully to the future of mathematical research.

The journey of integrating AI into mathematical research is just beginning, and the next generation of mathematicians will play a crucial role in shaping its direction. By embracing the opportunities while remaining mindful of the challenges, today's students can help ensure that AI serves to enhance rather than replace human mathematical insight and creativity.

Mathematical research has always been a collaborative endeavor, building upon the work of previous generations while pushing the boundaries of human knowledge. The integration of AI into this process represents a new form of collaboration – one that combines human creativity and insight with artificial intelligence capabilities. Success in this new paradigm requires not only technical skills but also wisdom, judgment, and a deep commitment to the values that make mathematical research meaningful.

The future of mathematics will be shaped by those who can navigate effectively between human intuition and artificial intelligence, between traditional mathematical reasoning and new computational capabilities, and between individual insight and collaborative discovery. This guide provides a foundation for that navigation, but the journey itself belongs to each researcher who chooses to explore the intersection of mathematics and artificial intelligence.

The core of mathematics remains unchanged. It is a discipline of rigor, creativity, and profound human thought. The most powerful tool you will ever have is your own mind. By learning to augment it with the technologies described in this guide—approaching them with skill, skepticism, and a strong ethical compass—you will be well-equipped to not only succeed in your studies, but to contribute to the future of mathematics itself.



Resources and Further Reading

A.1 AI TOOLS AND PLATFORMS

Large Language Models

- Claude (Anthropic): https://claude.ai Advanced reasoning and mathematical problem-solving
- **GPT-4 (OpenAI)**: https://openai.com Comprehensive language model with mathematical capabilities
- Gemini (Google): https://gemini.google.com Multimodal AI with mathematical reasoning
- Llama (Meta): https://llama.meta.com Open-source large language model

Computer Algebra Systems

- Mathematica: https://www.wolfram.com/mathematica/ Comprehensive symbolic computation
- Maple: https://www.maplesoft.com/products/maple/ Advanced mathematical software
- MATLAB: https://www.mathworks.com/products/matlab.html Numerical computing environment
- **SageMath**: https://www.sagemath.org Free open-source mathematics software
- **SymPy**: https://www.sympy.org Python library for symbolic mathematics

Proof Assistants and Formal Systems

- Lean: https://leanprover.github.io Modern theorem prover
- Coq: https://coq.inria.fr Formal proof management system
- Isabelle/HOL: https://isabelle.in.tum.de Generic proof assistant
- Agda: https://wiki.portal.chalmers.se/agda/ Dependently typed functional programming language

Specialized Mathematical AI Tools

- Wolfram Alpha: https://www.wolframalpha.com Computational knowledge engine
- **DeepMind AlphaGeometry**: Research tool for geometric theorem proving
- **OpenAI Codex**: AI system for code generation and mathematical computation

A.2 Educational Resources

Books on AI in Mathematics

- Bundy, A. (2023). *Automated Theorem Proving in Mathematics*. Cambridge University Press.
- Harrison, J. (2022). *Handbook of Practical Logic and Automated Reasoning*. Cambridge University Press.
- Wiedijk, F. (Ed.). (2021). The Seventeen Provers of the World. Springer.
- Paulson, L. C. (2020). *Logic and Computation: Interactive Proof with Cambridge LCF*. Cambridge University Press.

Online Courses and Tutorials

- Coursera: "Mathematics for Machine Learning" specialization
- edX: "Automated Reasoning" course series
- MIT OpenCourseWare: "Artificial Intelligence" and "Mathematics for Computer Science"
- Stanford Online: "Introduction to Mathematical Thinking"

A.3 Professional Organizations and Communities

Mathematical Societies

- American Mathematical Society (AMS): https://www.ams.org
- Mathematical Association of America (MAA): https://www.maa.org
- Society for Industrial and Applied Mathematics (SIAM): https:// www.siam.org
- International Mathematical Union (IMU): https://www.mathunion.org

AI and Computing Organizations

- Association for Computing Machinery (ACM): https://www.acm.org
- IEEE Computer Society: https://www.computer.org
- Association for the Advancement of Artificial Intelligence (AAAI): https://www.aaai.org

A.4 Conferences and Workshops

AI in Mathematics Conferences

- Conference on Automated Deduction (CADE)
- International Conference on Automated Reasoning (IJCAR)
- Conference on Intelligent Computer Mathematics (CICM)
- Workshop on AI for Mathematical Reasoning

General Mathematics Conferences

- Joint Mathematics Meetings (JMM)
- International Congress of Mathematicians (ICM)
- SIAM Annual Meeting

- [1] Mark J. Ablowitz and Peter A. Clarkson. *Solitons, Nonlinear Evolution Equations and Inverse Scattering*. Cambridge University Press, 2011.
- [2] Anthropic. Claude AI Assistant. AI-generated content within 'AI in Mathematics Research: A Comprehensive Guide for Senior Undergraduate Students', 2025. Conversation dated July 8, 2025. Claude Sonnet 4 model.
- [3] Jeremy Avigad. Mathematical reasoning and artificial intelligence. *Communications of the ACM*, 65(8):44–51, 2022.
- [4] Kshitij Bansal, Sarah Loos, Markus Rabe, Christian Szegedy, and Stewart Wilcox. Holist: An environment for machine learning of higher order logic theorem proving. In *International Conference on Machine Learning*, pages 454–463, 2019.
- [5] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.
- [6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020.
- [7] Alan Bundy. Automated Theorem Proving in Mathematics. Cambridge University Press, 2023.
- [8] François Chollet. On the measure of intelligence. arXiv preprint, 2019.
- [9] K. M. Collins, Albert Q. Jiang, Simon Frieder, L. Wong, M. Zilka, M. Lukasik, J. Ridder, D. Ranaldi, T. Salvatori, C. Sloper, E. O'Keefe, A. Engel, J. Y. Lee, and Mateja Jamnik. Structured, flexible, and robust: benchmarking and improving large language models towards more human-like behavior in out-of-distribution reasoning tasks. arXiv preprint, 2022.
- [10] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Cédric Rosset. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*, 2020.

- [11] Ernest Davis and Gary Marcus. The scope and limits of simulation in automated reasoning. *Journal of Automated Reasoning*, 67(2):1–25, 2023.
- [12] Leonardo de Moura et al. The Lean theorem prover (system description). In 25th International Conference on Automated Deduction (CADE-25), volume 9195 of Lecture Notes in Computer Science, pages 378–388. Springer, 2015.
- [13] Lawrence C. Evans. *Partial Differential Equations*. American Mathematical Society, 2nd edition, 2010.
- [14] Alhussein Fawzi, Matej Balog, et al. Discovering faster matrix multiplication algorithms with reinforcement learning, 2022.
- [15] Eitan First, Markus N. Rabe, Talia Ringer, and Yuriy Brun. Baldur: Whole-proof generation and repair with large language models. In Proceedings of the 2023 ACM SIGSOFT International Symposium on Software Testing and Analysis, pages 1–12, 2023.
- [16] Wendell H. Fleming and Raymond W. Rishel. Deterministic and Stochastic Optimal Control. Springer-Verlag, 2006.
- [17] Simon Frieder, Luca Pinchetti, Ryan-Rhys Griffiths, Tommaso Salvatori, M. Lukasik, P. C. Petersen, A. Chevalier, J. Berner, M. Ganea, A. T. Cemgil, and Mateja Jamnik. Mathematical capabilities of Chat-GPT. arXiv preprint, 2023.
- [18] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
- [19] W. T. Gowers and M. Ganesalingam. Can a computer recognize a mathematical proof? *Philosophical Transactions of the Royal Society A*, 380(2229):20210068, 2022.
- [20] John Harrison. *Handbook of Practical Logic and Automated Reasoning*. Cambridge University Press, 2022.
- [21] 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. In *Advances in Neural Information Processing Systems*, volume 34, pages 7893–7906, 2021.
- [22] Albert Q. Jiang, Sean Welleck, Jin Peng Zhou, Wenda Li, Jiacheng Liu, Mateja Jamnik, Guillaume Lample, and Yuhuai Wu. Draft, sketch, and prove: Guiding formal theorem proving with informal proofs. In *International Conference on Learning Representations*, 2023.
- [23] Andrej Karpathy. State of GPT. Talk at Microsoft Build Conference, 2023.

- [24] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman, Aristotelis Zhmoginov, Yuhuai Dronov, Dani Yogatama, Wojciech Fedus, Cédric Urban, Gergely van den Driessche, Jiaming Li, Da Zhou, Da Yan, Sean Welleck, Borja García, Yi Yuan, Jason Wei, Feiyang Niu, Laria Zou, Muge Smaili, Awa Kasirzadeh, Johnny Liao, Noah schärli, Wy Schalk, Josip Hacene, Hylary Soyer, Machel Bosma, and xiang Chen. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 3843–3857, 2022.
- [25] Wenda Li, Linyuan Yu, Yuhuai Wu, and Lawrence C. Paulson. Formal mathematics statement curriculum learning. In *International Conference on Learning Representations*, 2023.
- [26] Sarah Loos, Geoffrey Irving, Christian Szegedy, and Cezary Kaliszyk. Deep network guided proof search. In *International Conference on Logic for Programming, Artificial Intelligence, and Reasoning*, pages 85–105, 2017.
- [27] Maplesoft. Maple. Waterloo, Ontario: Maplesoft, a division of Waterloo Maple Inc., 2024.
- [28] Gary Marcus. Very preliminary notes on ChatGPT. Substack post, 2022.
- [29] M. Meadows and A. Freitas. Teaching large language models to reason with reinforcement learning. arXiv preprint, 2023.
- [30] James D. Murray. *Mathematical Biology I: An Introduction*. Springer-Verlag, 3rd edition, 2002.
- [31] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.
- [32] OpenAI. ChatGPT, 2024. Model: GPT-4. Accessed on: 2024-05-21.
- [33] OpenAI. Image generated by ChatGPT-40 based on user prompt. ChatGPT, July 2025.
- [34] Stanislas Polu and Ilya Sutskever. Generative language modeling for automated theorem proving. arXiv preprint, 2020.
- [35] Markus N. Rabe, Dennis Lee, Kshitij Bansal, and Christian Szegedy. Mathematical reasoning via self-supervised skip-tree training. In *International Conference on Learning Representations*, 2021.
- [36] Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. Physicsinformed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

- [37] David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing mathematical reasoning abilities of neural models. In *International Conference on Learning Representations*, 2019.
- [38] Steven H. Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. Westview Press, 2nd edition, 2014.
- [39] Animesh Thakur, Tsun-Hsuan Tsai, Aditya Grover, and Pieter Abbeel. Augmented neural ODEs. In *Advances in Neural Information Processing Systems*, volume 34, pages 3134–3145, 2021.
- [40] The MathWorks, Inc. MATLAB version R2024a, 2024.
- [41] The Sage Developers. SageMath, the Sage Mathematics Software System (Version 10.3), 2024.
- [42] Thang H. Trinh, Yuhuai Wu, Quoc V. Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625:476–482, 2024.
- [43] Josef Urban and Jiří Vyskočil. Theorem proving in large formal mathematics as an AI challenge. AI Magazine, 32(2):51–60, 2011.
- [44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008, 2017.
- [45] Lei Wang, Wenshan Xu, Yichong Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Annual Meeting of the Association for Computational Linguistics*, 2023.
- [46] Sean Welleck, Jiacheng Liu, Ronan Le Bras, Hannaneh Hajishirzi, Yejin Choi, and Kyunghyun Cho. NaturalProver: Grounded mathematical proof generation with language models. In *Advances in Neural Information Processing Systems*, volume 34, pages 4913–4927, 2021.
- [47] Wolfram Research, Inc. Mathematica, Version 14.0, 2024.
- [48] Yuhuai Wu, Markus N. Rabe, Wenda Li, Jimmy Ba, Roger B. Grosse, and Christian Szegedy. Autoformalization with large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 32353–32368, 2022.
- [49] Kaiyu Yang, Aidan Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan Mizrahi, and Anima Anandkumar. LeanDojo: Theorem proving with retrieval-augmented language models. arXiv preprint, 2023.

- [50] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yong Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. arXiv preprint, 2023.
- [51] Bernt Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Springer-Verlag, 6th edition, 2003.



Index

LATEX, 8, 16, 56, 66, 68, 70, 75, 81, 93

Abel-Ruffini theory, 5 Anthopic, 15 ARIMA, 41

backpropagation, 47 Bard, 16 Beamer, 76, 81 Minimal, 86 bisection method, 6 Bokeh, 66 Boyer-Moore theorem prover, 11 Brainstorm, 89 Brainstorming, 53

Chaining, 31 Chat GPT, 12 Chatbots, 15 Claude, 12, 15, 28 code generation, 6, 55 Collatz Conjecture, 44 combinatorics, 41, 89 complexity, 42 Computer algebra systems, 16 computer algebra systems, 11 Constraint specification, 29 Coq, 12, 20 Counterexamples, 53

DALL-E, 19 data analysis, 40 debugging, 78 deep learning, 12 differential equation ordinary, 36 differential equations partial, 37 Disclosure Statement, 94 Document Templates, 79 dynamical system, 45

eigenvalues, 53 Error handling, 29, 78 error handling, 56 Ethical disclosure, 93 Euler's formula, 45 Excel, 19 extrema, 6 Fibonacci sequence, 45 forecasts, 41 Formal Systems, 20 Four Color Theorem, 21 four-color theorem, 11 Fourier analysis, 45 Fourier Series, 82 Fourier space, 39 Gemini, 16 Generate script, 90 Goldbach conjecture, 58 golden ratio, 45 Google, 16 GPT, 15, 28 graph theory, 41 graphics, 16, 65 publication-quality, 55, 65 hallucination, 16 heat equation, 37 higher-order logic, 20 Hopf bifurcation, 45 image generation, 19 integer partitions, 89 interactive visualizations, 66 Intermediate Value Theorem, 6 Isabelle, 12, 20 Iterative refinement, 28 iterative refinement, 56 Kepler Conjecture, 21 Korteweg-de Vries equation, 39 Lagrange multipliers, 52 large language models, 1 Large language models, 15 Lean, 12, 20 LLM, 1 loss function, 38 Lotka-Volterra equations, 37 machine learning, 12

MACSYMA, 11, 13 Riemann Hypothesis, 28 Maple, 11, 17 Maplesoft, 17 Mathematica, 11, 16, 17, 70 MATLAB, 11, 17, 18 Matplotlib, 7, 20, 55, 65, 66, 68, 70, 92 Mayavi, 65 metaheuristics, 42 Microsoft Excel, 19 Midjourney, 19 Neural Network Training, 46 numerical integration, 57 NumPy, 17, 55 OpenAI, 15 optimization, 42, 45 AI-Assisted, 52 optimizers, 47 ordinary differential equation, 36 partial differential equations, 37 partitions integer, 89 phase plane, 37 Physics-Informed Neural Network, 37 PINN, 37 Plotly, 65, 66 portfolio optimization problem, 45 presentations, 76, 81 prime number distribution, 43 prime numbers, 43 problem solving, 35, 51 prompting, 45 prompts, 27 Proof Assistants, 20 Proof assistants, 62 Proof finding, 92 Python, 6, 17, 36, 55, 65, 70, 92 PyTorch, 38

raster graphics, 70

Rocq Prover, 20 role-playing, 29 root finding, 5 SageMath, 18 SciPy, 6, 17, 55 Seaborn, 65 secant method, 6 social network, 41 solitons, 39 spectral graph theory, 53 Square wave, 83 Stable Diffusion, 19 stock data, 41 symbolic computation, 1, 16 Symbolic integration, 63 SymPy, 17, 18, 55 Table and figure formatting, 76 Theorem environment, 76 theorem proving systems, 11 TikZ, 20 Tikz, 70 time series, 41 Traveling Salesman Problem, 42 validation, 56, 61 vector graphics, 70 Verification, 7 verification, 29, 61, 92 Verification and testing, 56 verification systems, 62 verification tools automated, 63 video generation, 19 Visualization, 7 visualization, 92 Three-dimensional, 66 visualizations, 65

Wolfram, 16 Wolfram Alpha, 16