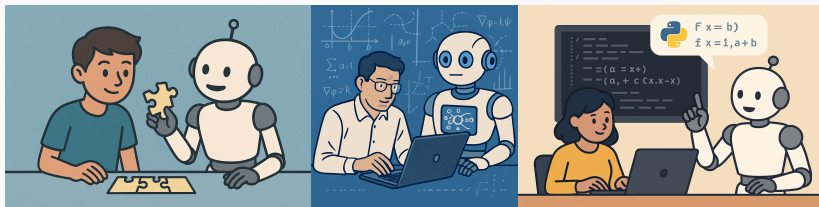


AI Literacy in Mathematics and Statistics

Tools, Techniques, and Responsible Use

Russell L. Herman, UNC Wilmington

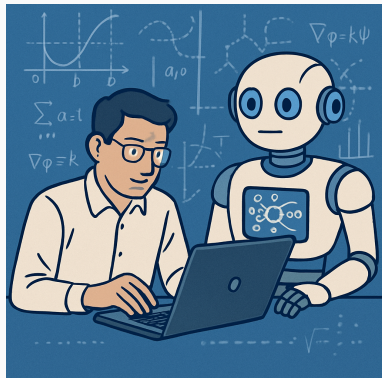


We introduce AI literacy for mathematics and statistics students, focusing on effective prompting, code generation, and verification. Through live demonstrations, students will learn to use AI tools strategically while recognizing their limitations. We'll explore common failure modes, ethical use in coursework, and best practices for citing AI assistance.

Emphasizing AI as a tool to support—not replace—mathematical thinking, the session offers practical guidance for integrating AI into communication and problem-solving. Students will leave with frameworks for responsible, transparent use of AI in academic work.

Today's Journey

1. Why AI literacy matters for mathematicians.
2. Basic AI interaction.
3. Strategic prompting techniques.
4. Code generation and verification.
5. When AI fails: Common pitfalls.
6. Ethics and institutional policies.



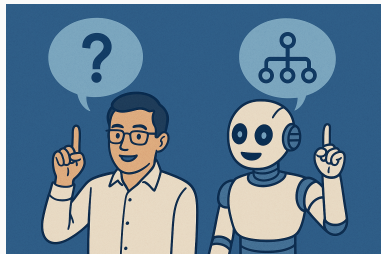
From *AI in Mathematics Research, A Comprehensive Guide for Senior Undergraduate*, August 2025.

<https://people.uncw.edu/hermanr/ai.htm>

Poll Question

How many of you have used AI for mathematical work?

- Never used it.
- Used for basic calculations.
- Used for code generation.
- Used for writing assistance.
- Used for research ideas.



Keep your answer in mind – we'll revisit this later.

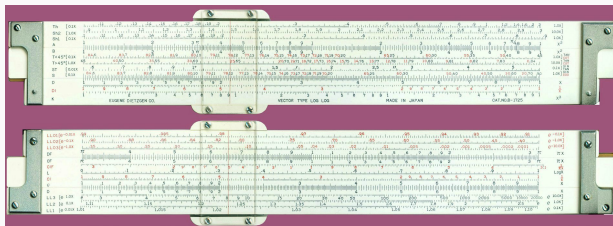
Past Revolutions Affecting Mathematics

Calculational Tools

- Counting sticks, knots,
- Abacus (soroban, suanpan).
- Slide Rules (1622).
- Electronic Calculators (1970).
- Personal Computers (1981).
- Computer Algebra Systems (CAS).

Information Access

- Clay, paper, scrolls,
- Printing Press (1440).
- Internet (1969).
- World Wide Web (1990).
- WebCrawler (1994).
- Wikipedia (2001).



The AI Revolution in Mathematics

In the Beginning:

- Alan Turing, “can machines think?” (1950).
- 1956 Dartmouth Summer Research Project on AI.
- FORTRAN, LISP, ELIZA (1st chatterbot),

Yesterday:

- Calculators, CAS systems.
- Computational assistance.
- Predetermined algorithms.

Today:

- Natural language interaction.
- Code generation.
- Explanation and tutoring.
- Pattern recognition.

Senior and Graduate Mathematics Seminars

Key Insight:

AI doesn't replace mathematical thinking – it amplifies it.

Chatterbots - History

- ELIZA 1968 mimics human conversation.
- Parry 1972 imitated a patient with schizophrenia.
- Jabberwocky 1988 simulate natural human conversation.
- Dr. Sbaitso 1992 full voice operated.
- A.L.I.C.E. 1995 simulated chatting with a real person over the Internet.
- SmartChild 2001 a precursor of Siri.
- Siri 2010 intelligent personal assistant.
- Google Now 2012.
- Cortana 2014.
- Alexa 2014

```
Welcome to
          EEEEE LL   IIII ZZZZZ AAAAA
          EE    LL   II    ZZ  AA  AA
          EEEEE LL   II    ZZZ AAAAAA
          EE    LL   II    ZZ  AA  AA
          EEEEE LLLLL IIII ZZZZZ AA  AA

Eliza is a mock Rogerian psychotherapist.
The original program was described by Joseph Weizenbaum in 1966.
This implementation by Norbert Landsteiner 2005.

ELIZA: Is something troubling you ?
YOU:  Men are all alike.
ELIZA: What is the connection, do you suppose ?
YOU:  They're always bugging us about something or other.
ELIZA: Can you think of a specific example ?
YOU:  Well, my boyfriend made me come here.
ELIZA: Is it important to you that your boyfriend made you come here ?
YOU:  He says I'm depressed much of the time.
ELIZA: I am sorry to hear that you are depressed.
YOU:  It's true. I am unhappy.
ELIZA: Can you explain what made you unhappy ?
YOU:
```

```
Sound Blaster          DR SBAITSO          version 2.20
(c) Copyright Creative Labs, Inc. 1992, all rights reserved

Please enter your name ...Dave
HELLO DAVE, MY NAME IS DOCTOR SBAITSO.

I AM HERE TO HELP YOU.
SAY WHATEVER IS IN YOUR MIND FREELY.
OUR CONVERSATION WILL BE KEPT IN STRICT CONFIDENCE.
MEMORY CONTENTS WILL BE WIPED OFF AFTER YOU LEAVE.

SO, TELL ME ABOUT YOUR PROBLEMS.
>
```

AI Tools for Mathematics and Statistics (Part 1)

General Purpose AI:

- ChatGPT: <https://chat.openai.com>
- Claude: <https://claude.ai>
- Gemini: <https://gemini.google.com>
- Grok: <https://grok.com/>
- Copilot: <https://copilot.microsoft.com>

Specialized Math AI:

- Wolfram Alpha: <https://www.wolframalpha.com>
- Photomath: <https://photomath.com>
- Symbolab: <https://www.symbolab.com>

Research & Analysis:

- Perplexity: <https://www.perplexity.ai>
- Consensus: <https://consensus.app>
- Elicit: <https://elicit.org>
- Semantic Scholar: <https://www.semanticscholar.org>



ChatGPT



Claude



Gemini



Grok



Copilot

AI Tools for Mathematics and Statistics (Part 2)

Code Generation:

- GitHub Copilot: <https://github.com/features/copilot>
- Replit: <https://replit.com>
- Cursor: <https://cursor.sh>
- Tabnine: <https://www.tabnine.com>

Document Preparation:

- Overleaf AI: <https://www.overleaf.com>
- Underleaf <https://app.underleaf.ai/>
- Notion AI: <https://www.notion.so>
- Grammarly: <https://grammarly.com>

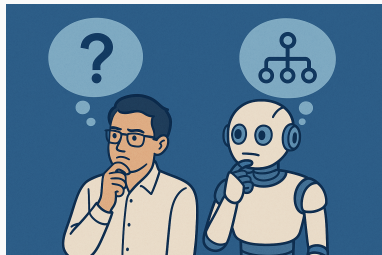
Visualization & Computation:

- Observable: <https://observablehq.com>
- Jupyter AI: <https://jupyter.org>
- Colab: <https://colab.research.google.com>

Using AI Tools

Getting Started:

- Start with free tiers.
- Compare outputs across tools.
- Always verify AI-generated content.
- Document your AI assistance.



Note: Check UNCW's policies and your course policies before using these tools for coursework.

The image was generated in ChatGPT using the prompt: I want a version of the characters in the title page which shows them thinking before they begin to use AI tools so I can add it to a beamer slide.

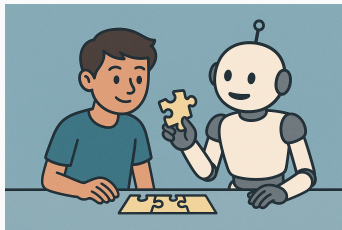
First Steps with AI

Let's solve together:

“Find the critical points of $f(x) = x^3 - 6x^2 + 9x + 2$.”

We'll try different prompting approaches:

1. Basic request.
2. Request with verification.
3. Request with explanation.
4. Request with visualization.



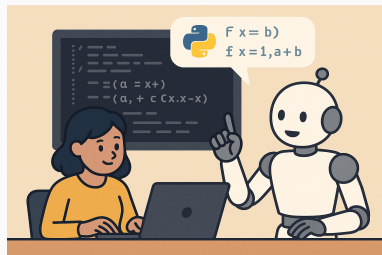
What We Just Observed

The AI provided:

- Correct derivative calculation.
- Critical point identification.
- Python code for verification.
- Clear explanation of steps.

But notice:

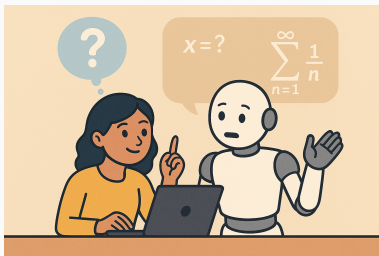
- We had to direct the process.
- We needed to verify the results.
- Quality depended on our prompts.



Anatomy of Effective Prompts

Four Key Elements:

1. **Context:** Set the mathematical domain.
2. **Task:** Specify what you want.
3. **Constraints:** Define limits and requirements.
4. **Format:** Specify output style.



Prompt Evolution: Collatz Conjecture

Weak:

“Explain the Collatz Conjecture.”

Better:

“Explain the Collatz Conjecture to calculus students, including why it’s unsolved.”

Strategic:

“Create three different visualizations of the Collatz Conjecture that would help calculus students understand both the pattern and why it’s difficult to prove. Include Python code.”

Code Output - Collatz Conjecture

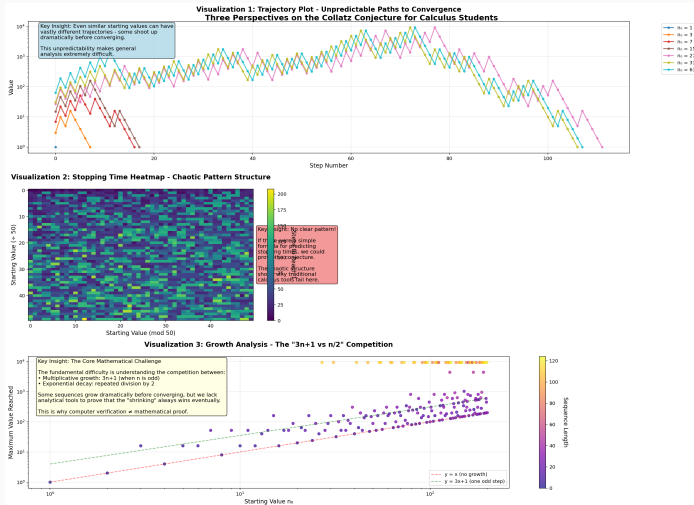


Figure 1: Output from Python Code.

Example Research Question Development:

1. "List 5 accessible research questions in graph theory for undergraduates."
2. "Take the second question. Outline a 3-page research summary structure."
3. "For the first section, draft 200 words of background with 3 key references."
4. "Generate LaTeX code for a theorem statement related to this question."

Prompt Chaining: Building Complex Results

Example Research Question Development:

1. “List 5 accessible research questions in graph theory for undergraduates.”
2. “Take the second question. Outline a 3-page research summary structure.”
3. “For the first section, draft 200 words of background with 3 key references.”
4. “Generate LaTeX code for a theorem statement related to this question.”

Each step builds on the previous – you maintain control of direction.

Sample Activity:

Write a strategic prompt for one of these scenarios:

- Explain eigenvalues to someone who understands matrices.
- Generate code to visualize a parametric surface.
- Create practice problems for integration by parts.
- Outline a presentation on Bayes' theorem applications.

Include: context, task, constraints, format.

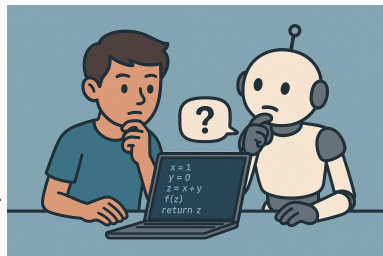
Code Generation and Verification

Challenge: Solve and visualize the ODE:

$$\frac{dy}{dt} = -2ty, \quad y(0) = 1.$$

We want:

- Numerical solution using Python.
- Analytical solution for comparison.
- Publication-quality plot.
- Verification that our solution is correct.



Generated Code Structure

```
1 import numpy as np
2 from scipy.integrate import solve_ivp
3 import matplotlib.pyplot as plt
4
5 def ode_function(t, y):
6     return -2 * t * y[0]
7
8 # Solve numerically
9 sol = solve_ivp(ode_function, [0, 2], [1], dense_output=True
10 )
11
12 # Analytical solution:  $y = \exp(-t^2)$ 
13 t_analytical = np.linspace(0, 2, 100)
14 y_analytical = np.exp(-t_analytical**2)
15
16 # Plot comparison
17 plt.plot(sol.t, sol.y[0], 'b-', label='Numerical')
18 plt.plot(t_analytical, y_analytical, 'r--', label='
    Analytical')
```


Code Output - Solution of $\frac{dy}{dt} = -2ty, \quad y(0) = 1.$

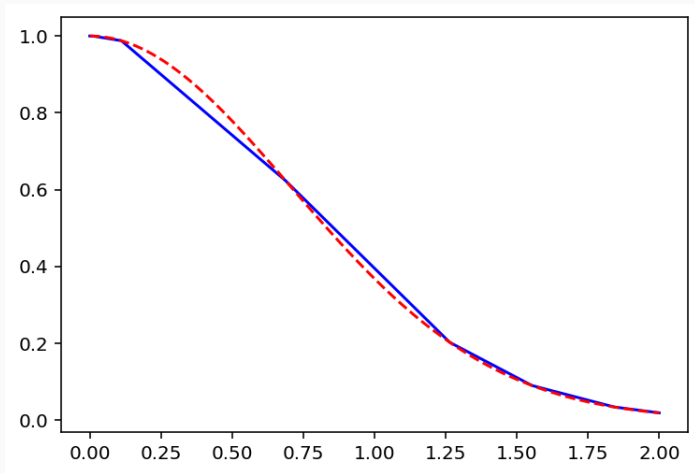


Figure 2: Solution from Python Code.

Always check AI-generated code:

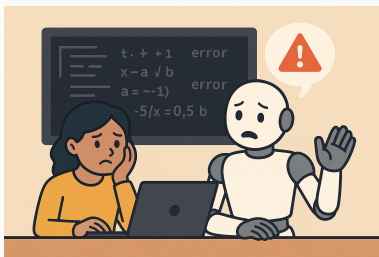
- Does it run without errors?
- Are the results mathematically sensible?
- Does it match known solutions?
- Is the approach appropriate for the problem?

In our example:

- Numerical and analytical solutions should overlap.
- Initial condition $y(0) = 1$ should be satisfied.
- Solution should decay as expected.

When AI Goes Wrong: Common Failures

1. **Hallucinated citations:** Fake papers, wrong authors.
2. **Mathematical errors:** Incorrect formulae, flawed logic.
3. **Overconfidence:** Certain tone with wrong answers.
4. **Missing context:** Technically correct but inappropriate.
5. **Inconsistency:** Contradictory statements within response.



Case Study: AI Mathematical Error

Prompt: “Prove that the derivative of $\sin(x^2)$ is $\cos(x^2)$.”

AI might respond: “The derivative of $\sin(x^2)$ is indeed $\cos(x^2)$ by the basic derivative rule for sine functions ...”

Case Study: AI Mathematical Error

Prompt: “Prove that the derivative of $\sin(x^2)$ is $\cos(x^2)$.”

AI might respond: “The derivative of $\sin(x^2)$ is indeed $\cos(x^2)$ by the basic derivative rule for sine functions ...”

The Error:

- Correct answer: $\frac{d}{dx} \sin(x^2) = 2x \cos(x^2)$.
- AI forgot the chain rule.
- Confident presentation masked the mistake.

Red Flags: When to Be Suspicious

- **No uncertainty:** Real math often involves “it depends” or conditions.
- **Perfect citations:** Especially for recent or obscure work.
- **Circular reasoning:** Assumptions that prove themselves.
- **Missing steps:** “It’s easy to show that...” without showing.
- **Contradictory information:** Within the same response.

Your defense: Always verify, especially for important results.

AI and Mathematical Proofs: Reality Check

What AI can do:

- Verify steps in standard proofs.
- Suggest proof strategies and approaches.
- Help with routine calculations within proofs.
- Generate proof outlines for familiar theorems.

What AI cannot reliably do:

- Create novel, rigorous proofs.
- Handle complex multi-step arguments.
- Provide formal verification.
- Replace mathematical insight and creativity.

Key takeaway: Use AI for proof exploration and verification, never as the final authority on mathematical correctness.

AI and Mathematical Proofs: Setting Expectations

The Big Question:

Can AI generate rigorous mathematical proofs beyond textbook examples?

Short Answer: Limited success with significant gaps.

What we'll explore:

- Where AI succeeds with proofs.
- Common failure modes.
- Live demonstration of AI proof attempts.
- Guidelines for proof-related AI use.

AI Proof Capabilities: The Spectrum

AI can handle:

- Standard calculus proofs.
- Basic algebraic arguments.
- Routine geometric proofs.
- Verification of given steps.
- Proof strategy suggestions.
- Converting informal arguments to formal structure.

AI struggles with:

- Novel proof techniques.
- Multi-lemma constructions.
- Creative mathematical insights.
- Formal verification systems.
- Research-level theorems.
- Recognizing when proofs are impossible.

Note: , OpenAI and Google DeepMind won a gold medal at the 2025 International Math Olympiad (IMO), solving five out of six problems.

However, LLMs predict what text comes next in a string of words based on what's in its dataset. They may produce several “proofs” and “judges” what looks best. It may not be reproducible or verifiable.

Prompt: “Prove that the sum of two even integers is even. Show each step clearly.”

Expected AI Response:

- Let $a = 2m$ and $b = 2n$ for integers m, n .
- Then $a + b = 2m + 2n = 2(m + n)$.
- Since $m + n$ is an integer, $a + b$ is even.

Assessment: Correct structure, clear logic, appropriate level.

Live Test: Intermediate Proof

Prompt: “Prove that $\sqrt{2}$ is irrational using proof by contradiction.”

What AI typically provides:

- Assume $\sqrt{2} = \frac{p}{q}$ in lowest terms.
- Square both sides: $2 = \frac{p^2}{q^2}$, so $2q^2 = p^2$.
- Therefore p^2 is even, which means p is even.
- Let $p = 2k$, then $2q^2 = 4k^2$, so $q^2 = 2k^2$.
- This means q is also even, contradicting lowest terms.

Assessment: Usually correct, but may skip justification steps.

Dangerous Prompt: “Prove that there are infinitely many twin primes.”

What AI might incorrectly claim:

- “This follows from Dirichlet's theorem ...”
- “We can adapt Euclid's proof for regular primes ...”
- Confident tone despite this being an open problem.

Red Flags:

- No mention that this is unsolved.
- Overconfident language.
- Handwaving at crucial steps.

Reality Check: This is the Twin Prime Conjecture - still open!

AI Tool Performance Ranking for Proofs

Based on mathematical rigor and self-awareness:

1. Claude (Anthropic)

- More cautious about unsolved problems.
- Better at acknowledging uncertainty.
- Clearer about proof limitations.

2. GPT-4 (OpenAI)

- Strong at proof sketches and strategies.
- Can be overconfident with false claims.
- Good at explaining proof techniques.

3. Specialized Tools

- Lean + AI: Emerging formal verification.
- Limited scope but higher reliability.

Common Proof-Related Failures

Pattern Recognition: Know these warning signs.

1. **Circular reasoning:** Assuming what needs to be proved.
2. **Missing cases:** Incomplete case analysis.
3. **Unjustified steps:** “Clearly” or “Obviously” without proof.
4. **False confidence:** Certain tone about open problems.
5. **Technique misapplication:** Using wrong proof methods.
6. **Logical gaps:** Steps that don’t actually follow.

Your job: Catch these errors through careful verification.

Guidelines for AI-Assisted Proof Work

Safe practices:

- Use AI for proof *strategies* and *structure*.
- Verify every logical step independently.
- Cross-check with authoritative sources.
- Ask AI to *explain* rather than just *prove*.
- Test AI claims with simple examples.

Dangerous practices:

- Accepting AI proofs without verification.
- Using AI proofs for homework/exams without permission.
- Assuming AI knows current research status.
- Trusting complex multi-step arguments blindly.

Better Prompts for Proof Work

Instead of: “Prove that [statement]”

Try these approaches:

Strategy-focused:

“What are three different approaches to proving [statement]? Explain when each might be appropriate.”

Verification-focused:

“Check this proof step: [statement]. Is the logic sound? What might be missing?”

Exploration-focused:

“What lemmas would be needed to prove [statement]? Are there any known obstacles?”

Educational-focused:

“Explain the key insight behind the proof of [theorem]. What makes it work?”

The Bottom Line on AI and Proofs

Current Reality:

- AI can help with proof *exploration* and *understanding*.
- AI cannot reliably generate rigorous proofs independently.
- Human verification and insight remain essential.

Best Use Cases:

- Learning proof techniques through examples.
- Getting unstuck on proof strategies.
- Checking your reasoning for obvious errors.
- Understanding published proofs.

Remember: Mathematical truth requires human judgment, not just computational power.

Real Costs:

- API fees: \$0.01-\$0.10 per interaction.
- Time spent prompting and verifying.
- Environmental impact of model training/inference.
- Risk of developing dependence.

When AI Saves Time:

- Code generation and debugging.
- LaTeX formatting and syntax.
- Literature search and summarization.
- Explanation of unfamiliar concepts.

Rate these scenarios (A = Always OK, B = Sometimes OK, C = Never OK):

1. Using AI to check your calculus homework.
2. Having AI write your research paper introduction.
3. Using AI to generate data for a statistics project.
4. AI helping debug your programming assignment.
5. AI solving exam problems during a test.

Rate these scenarios (A = Always OK, B = Sometimes OK, C = Never OK):

1. Using AI to check your calculus homework.
2. Having AI write your research paper introduction.
3. Using AI to generate data for a statistics project.
4. AI helping debug your programming assignment.
5. AI solving exam problems during a test.

The key question: Are you learning or outsourcing?

Generally Acceptable:

- Explanation of concepts you're learning.
- Code syntax help and debugging assistance.
- LaTeX formatting and document preparation.
- Literature search and citation formatting.

Requires Disclosure:

- Substantial text generation or revision.
- Problem-solving strategies or solutions.
- Data analysis methods or interpretation.
- Any content that becomes part of your submission.

Example disclosure statement:

“The Python code for numerical integration (Section 3.2) was initially generated using ChatGPT-4 and subsequently verified and modified by the author. The AI provided the basic structure for the `scipy.integrate` approach, while parameter selection and error analysis were performed independently.”

Key elements: What AI did, what you did, how you verified it.

Your AI Literacy Action Plan

This Week:

1. Choose one AI tool and create an account.
2. Practice 3 different types of prompts.
3. Verify AI output using traditional methods.

This Month:

1. Use AI for a non-critical project.
2. Document what works and what fails.
3. Discuss policies with your instructors.

This Semester:

1. Develop your personal ethical framework.
2. Build prompt libraries for common tasks.
3. Share learnings with classmates.

Resources and Tools

AI Platforms:

- ChatGPT (OpenAI): General purpose, strong code generation
- Claude (Anthropic): Mathematical reasoning, ethical focus.
- Gemini (Google): Multimodal, integrated with Google tools.
- Grok (xAI): specializes in real-time data, social media analysis.

Mathematical Tools:

- Wolfram Alpha: Computational engine with AI integration.
- SageMath: Open-source mathematics software.
- DeepXDE: Physics-informed neural networks.

Learning Resources:

- Teaching Naked Prompts: <https://teachingnaked.com/prompts/>
- Anthropic Prompt Engineering: <https://docs.anthropic.com/en/docs/build-with-claude/prompt-engineering/overview>

How has your perspective changed?

Think about:

- What new applications do you see for your coursework?
- What concerns do you have about responsible use?
- What would you like to try first?

Discussion: Share one insight with a neighbor.

Key Takeaways

1. **You are the director:** AI responds to your guidance.
2. **Verification is mandatory:** Never trust without checking.
3. **Transparency builds trust:** Document AI assistance appropriately.
4. **Ethics vary by context:** Learning vs. evaluation vs. research.
5. **Practice develops judgment:** Start with low-stakes experiments.

Remember: AI is a powerful amplifier of human mathematical thinking, not a replacement for it.

Questions and Discussion

What questions do you have about integrating AI into your mathematical work?

What policies should we establish as a learning community?

What would you like to explore first?

