Artificial Intelligence and Mathematics

From Neural Networks to Computer-Assisted Proofs

AI & AL

ITI "E. Torricelli - G. Tomasi di Lampedusa"

April 2026

Prompt for this presentation

I am a math teacher at a technical high school (upper secondary school) and I would like to give a lesson on the use of AI in mathematics, particularly regarding computer-assisted proofs. This lesson is aimed at 12th and 13th grade students who already have a background in computer science and follow a rigorous mathematics curriculum.
I would like to start with a brief general introduction to AI (both current and potential) and its connection to mathematics, also from a formal perspective related to its development, not only in terms of applications. I would also like to explore the reverse process: what AI can do for mathematics, especially in the context of computer-assisted proofs. For example, recalling the role of computational tools in the Four Color Theorem, the Classification of Finite Simple Groups, or Knuth's latest graph classification results.
It would also be useful to include some final exercises on how to use AI to prove statements, working through some well-known ones and approaching the analysis of a few still-unsolved (very elementary) conjectures.
It would be great to have the final output as a presentation in English, also available in LaTeX format.

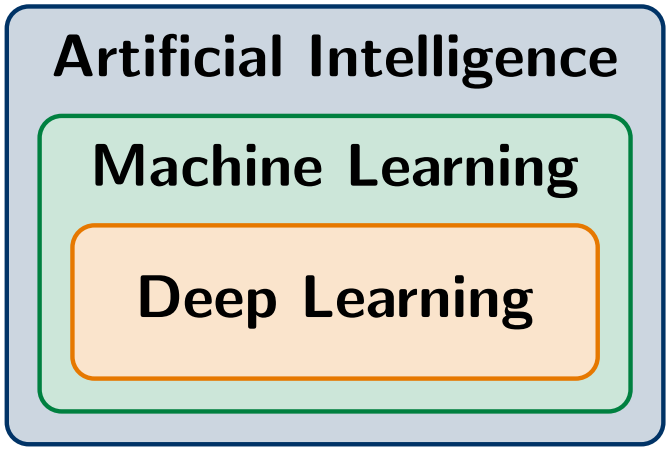
Lecture Outline

What is Artificial Intelligence?

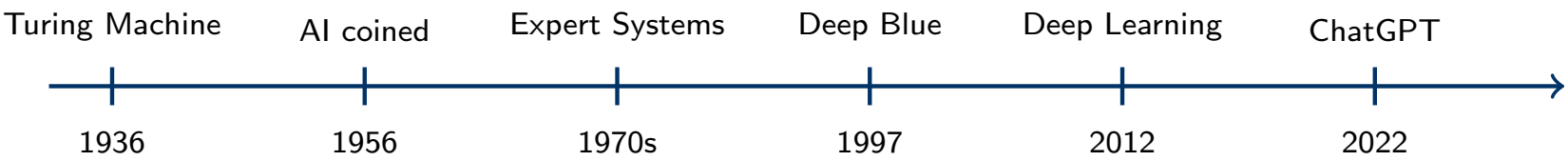
The Big Picture

Artificial Intelligence encompasses systems that perform tasks typically requiring human intelligence:

- **Learning** from data and experience
- **Reasoning** through logical inference
- **Problem-solving** in complex domains
- **Pattern recognition** in large datasets



Brief Historical Timeline



Key Insight

AI progress has always been **intertwined with mathematics**: from Turing's foundational theory to modern optimization algorithms.

How Modern AI Works: Neural Networks

A **neural network** approximates functions through layers of simple operations:

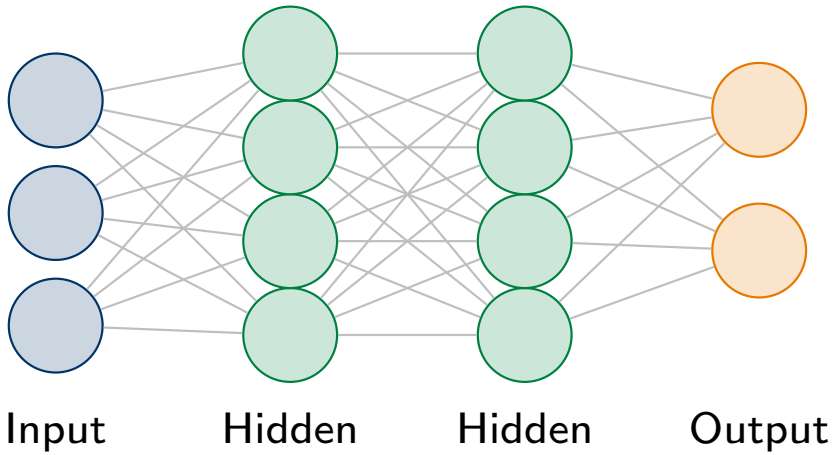
Mathematical Foundation

For input $\mathbf{x}$, output $\mathbf{y}$:

$$\mathbf{y} = f_n \circ f_{n-1} \circ \cdots \circ f_1(\mathbf{x})$$

Each layer:

$$f_i(\mathbf{z}) = \sigma(W_i \mathbf{z} + \mathbf{b}_i)$$



Where σ is a nonlinear **activation function** (e.g., ReLU: $\sigma(x) = \max(0, x)$)

The Universal Approximation Theorem

Theorem (Cybenko, 1989; Hornik, 1991)

A feedforward neural network with a single hidden layer containing a finite number of neurons can approximate any continuous function on compact subsets of $\mathbb{R}^n$, under mild assumptions on the activation function.

Formally: For any continuous $f : [0,1]^n \to \mathbb{R}$ and any $\varepsilon > 0$, there exist $N \in \mathbb{N}$, weights $w_i, \alpha_i \in \mathbb{R}$, and biases $b_i \in \mathbb{R}$ such that:

$$\left| f(\mathbf{x}) - \sum_{i=1}^{N} \alpha_i \sigma\left(\mathbf{w}_i \cdot \mathbf{x} + b_i\right) \right| < \varepsilon$$

Important Caveat

The theorem guarantees *existence*, not efficient *construction* or *learning*.

Large Language Models: The Transformer Architecture

Modern AI chatbots use **Transformers**, which rely on the **attention mechanism**:

Scaled Dot-Product Attention

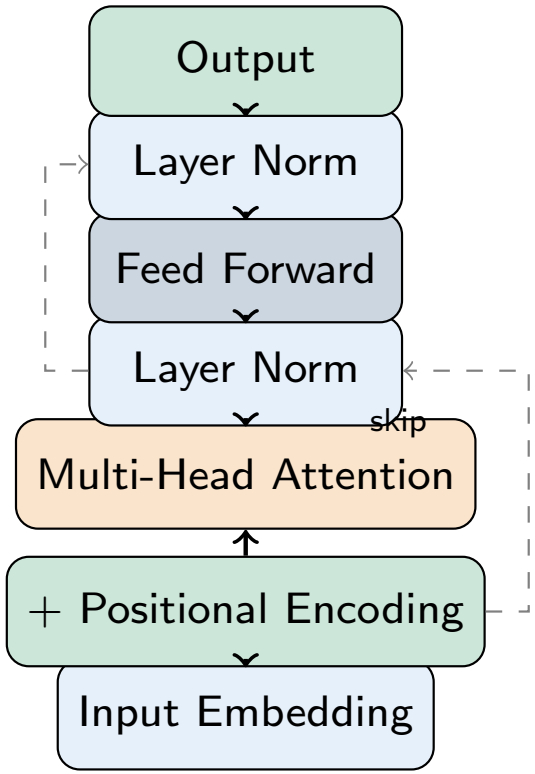
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Where:

- Q = Query matrix
- K = Key matrix
- V = Value matrix
- d_k = dimension of keys

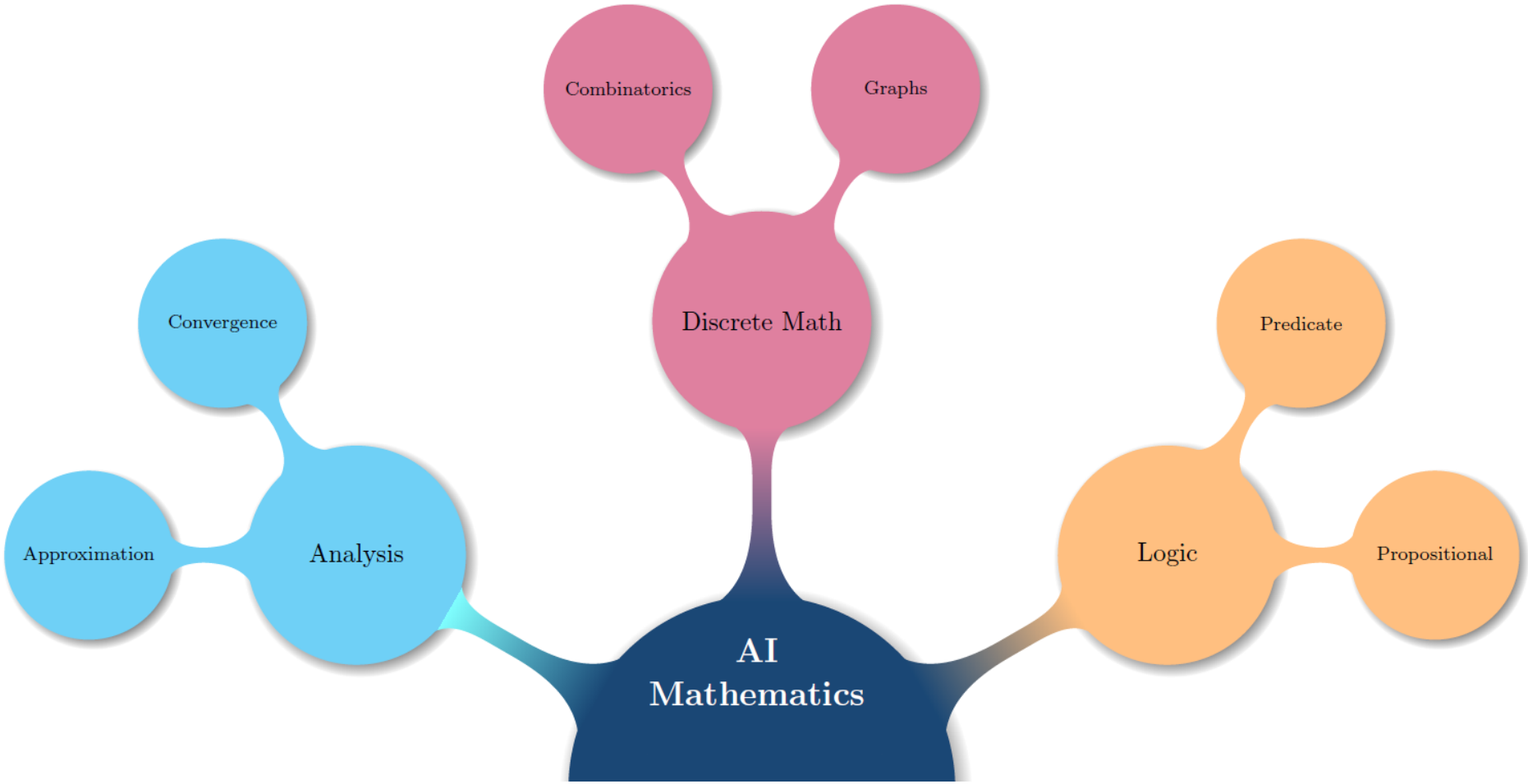
The **softmax** converts scores to probabilities:

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

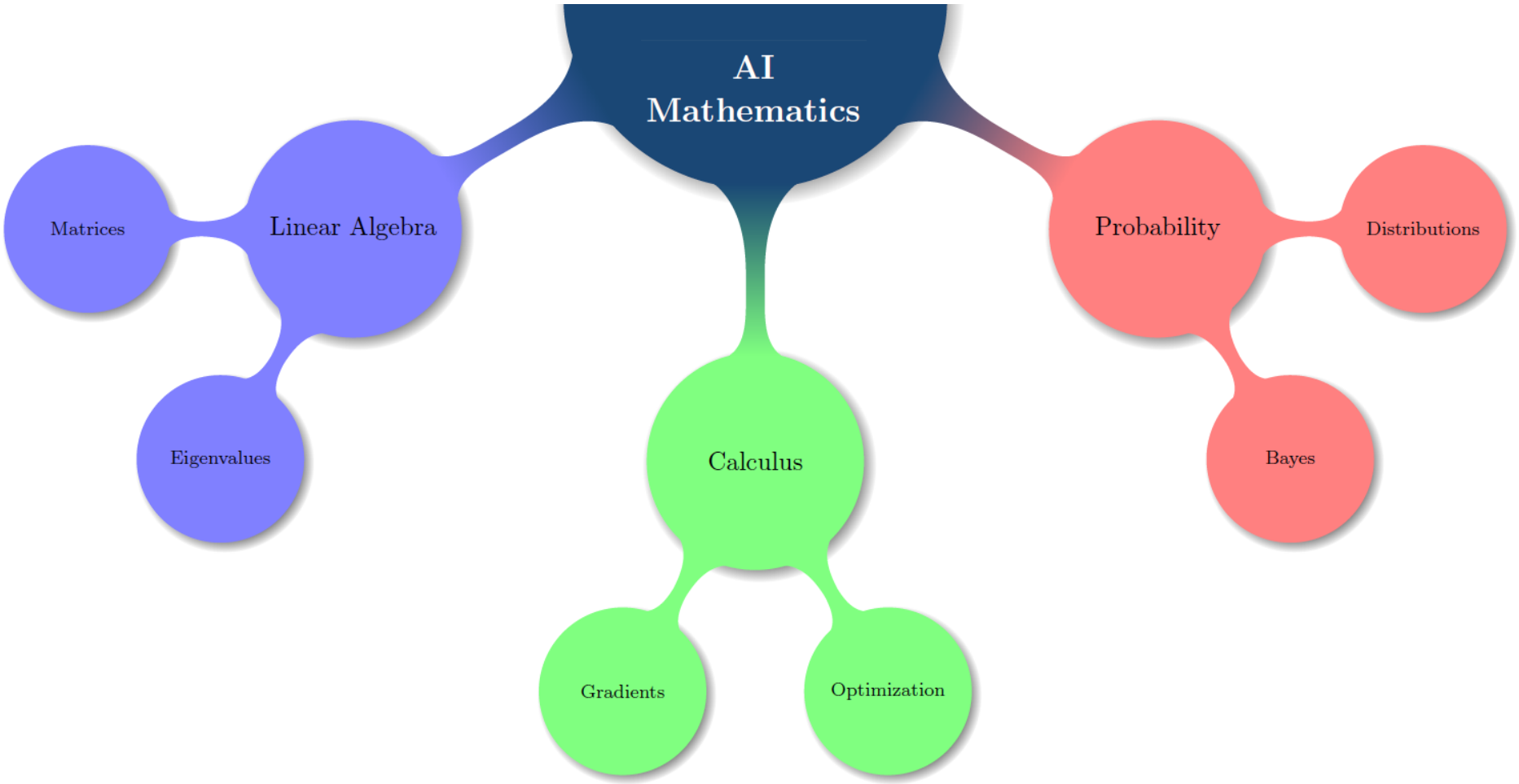


Mathematics as the Foundation of AI

The Mathematical DNA of Artificial Intelligence



The Mathematical DNA of Artificial Intelligence



Gradient Descent: Learning as Optimization

Problem: Find weights $\mathbf{w}$ that minimize a loss function $L(\mathbf{w})$.

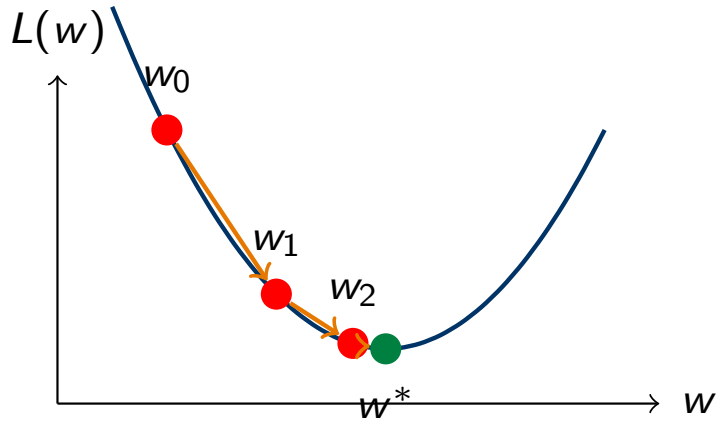
Gradient Descent Update Rule

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla L(\mathbf{w}_t)$$

Where:

- η = learning rate (step size)
- ∇L = gradient of loss function

Key insight: The gradient points in the direction of steepest *ascent*; we go opposite.



Connection to Calculus

Recall: $\nabla L = \left(\frac{\partial L}{\partial w_1}, \ldots, \frac{\partial L}{\partial w_n} \right)$

Backpropagation: The Chain Rule at Scale

To train neural networks, we need gradients of the loss with respect to *every* weight.

Chain Rule (Multivariable)

If $y = f(g(x))$, then:

$$\frac{dy}{dx} = \frac{dy}{dg} \cdot \frac{dg}{dx}$$

Example: For a two-layer network with loss L:

$$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h} \cdot \frac{\partial h}{\partial W_1}$$

Computational Graph

Backpropagation efficiently computes all gradients in $O(n)$ time by traversing the computation graph backwards.

Why Linear Algebra is Essential

Core operations in AI:

- **Matrix multiplication**: $Y = WX + B$
  - Every layer transformation
  - Attention computation
- **Eigendecomposition**: $A = Q\Lambda Q^{-1}$
  - Principal Component Analysis
  - Understanding network dynamics
- **SVD**: $A = U\Sigma V^T$
  - Dimensionality reduction
  - Model compression

Dimension Check

If $X \in \mathbb{R}^{n \times m}$ (n samples, m features)
and $W \in \mathbb{R}^{k \times m}$ (k neurons),
then $Y = WX^T \in \mathbb{R}^{k \times n}$

GPT-4 Scale

Estimated parameters:
$\sim 1.7 \times 10^{12}$
Matrix operations:
$\sim 10^{15}$/inference

Computer-Assisted Proofs

The Traditional View of Mathematical Proof

> *"A proof is a sequence of logical deductions from axioms and previously established statements that culminates in the proposition in question."*
>
> — Traditional definition

Classical requirements:

- Human-verifiable
- Surveyable (can be checked in finite time)
- Follows accepted logical rules

Emerging challenges:

- Proofs too long for humans
- Computational verification
- Collaborative mega-proofs

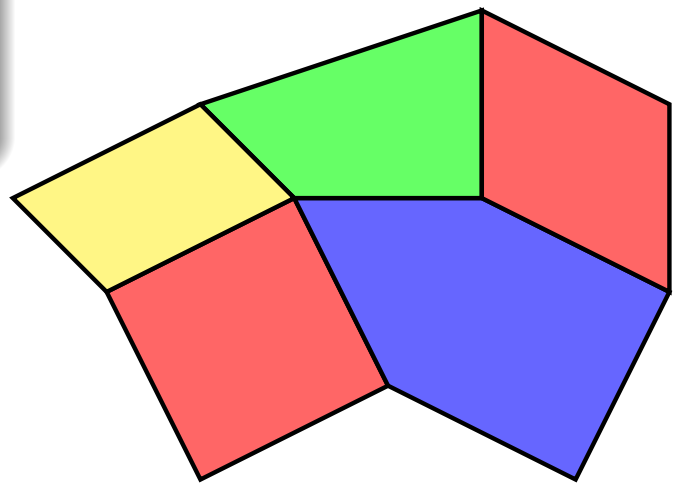
The Question

Is a computer-verified proof a "real" proof?

Landmark: The Four Color Theorem (1976)

Theorem (Appel & Haken, 1976)

Every planar map can be colored with at most four colors such that no two adjacent regions share the same color.



Four colors suffice!

The proof strategy:

1. Reduce to 1,936 "unavoidable" configurations
2. Prove each is "reducible" (can be simplified)
3. Computer checks all configurations

Computation: ∼1,200 hours on IBM 360

2005: Formally Verified

Georges Gonthier verified the theorem in Coq proof assistant. [gonthier2008formal]

The Classification of Finite Simple Groups

The Enormous Theorem

Every finite simple group is isomorphic to one of:

- A cyclic group $\mathbb{Z}_p$ of prime order
- An alternating group A_n for $n \geq 5$
- A group of Lie type
- One of 26 sporadic groups

The proof:

- Approximately **10,000+ pages** across hundreds of papers
- Over 100 mathematicians contributed (1955–2004)
- Daniel Gorenstein coordinated the effort
- Still being simplified and verified today

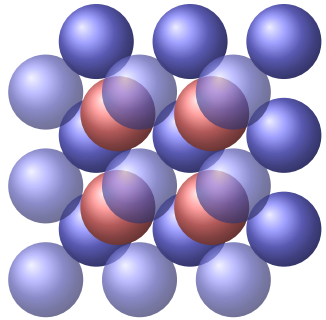
Computer's Role

Computations verified properties of sporadic groups (especially the Monster group with order $\approx 8 \times 10^{53}$).

The Kepler Conjecture: From Hales to Formal Proof

Kepler Conjecture (1611)

The densest packing of spheres in 3D is the face-centered cubic arrangement, with density $\frac{\pi}{3\sqrt{2}} \approx 0.7405$.



Timeline:

- **1998**: Thomas Hales announces proof
- **2003**: Referees "99% certain" after 4 years
- **2014**: Formal verification completed (Flyspeck project)
- Used **HOL Light** proof assistant

Lesson

Human proof took 4 years to review; formal verification provided certainty.

Boolean Pythagorean Triples Problem (2016)

Problem

Can the integers $\{1, 2, 3, \ldots\}$ be colored with two colors such that no Pythagorean triple (a, b, c) with $a^2 + b^2 = c^2$ is monochromatic?

Result (Heule, Kullmann, Marek):

- $\{1, \ldots, 7824\}$: **YES** — can be 2-colored
- $\{1, \ldots, 7825\}$: **NO** — impossible!

The proof:

- SAT solver computation
- **200 terabytes** of proof data
- Compressed to 68 GB
- 4 CPU-years of computation

Example Triple

$(3, 4, 5)$: $9 + 16 = 25$

If 3 and 4 are both red, then 5 must be blue.

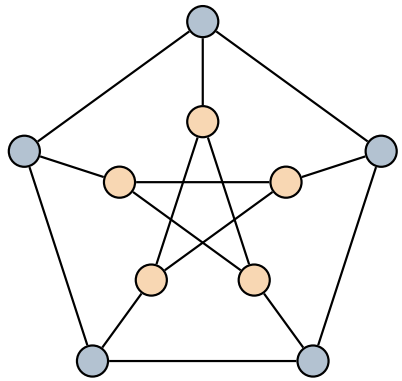
The coloring constraints propagate...

Donald Knuth and Graph Theory (2023–2025)

Context: In volumes of *The Art of Computer Programming*, Knuth has extensively used computational verification.

Recent work on graph enumeration and classification:

- Counting non-isomorphic graphs
- SAT-based verification of combinatorial identities
- Exhaustive computer search with proof certificates



Petersen Graph

Knuth's Philosophy

"I believe that the best way to understand something is to program it."

Proof Assistants and Formal Verification

What is a Proof Assistant?

Definition

A **proof assistant** (or interactive theorem prover) is software that helps construct mathematical proofs in a formal language that can be mechanically verified.

Key properties:

- Proofs are **machine-checkable**
- Every step is verified
- Cannot skip or hand-wave
- Trusted kernel is small

Major systems:

- **Lean** (Microsoft Research)
- **Coq** (INRIA, France)
- **Isabelle** (Cambridge/Munich)
- **HOL Light** (Harrison)

The Curry-Howard Correspondence

Proofs $\equiv$ Programs; Propositions $\equiv$ Types

A proof of $P \to Q$ is a function that takes evidence of P and returns evidence of Q.

Introduction to Lean

Lean is both a programming language and a proof assistant based on **dependent type theory**.

Basic Syntax

```
-- Define a theorem
theorem add_comm (a b : Nat) :
  a + b = b + a := by
  induction a with
  | zero => simp
  | succ n ih =>
    simp [Nat.succ_add, ih]
```

Key concepts:

- `theorem`: declares a statement
- `:=`: provides the proof
- `by`: enters tactic mode
- `induction`: proof strategy
- `simp`: simplification tactic

Mathlib

The **Mathlib** library contains 100,000+ theorems, covering undergraduate and graduate mathematics.

Lean Example: Basic Logic

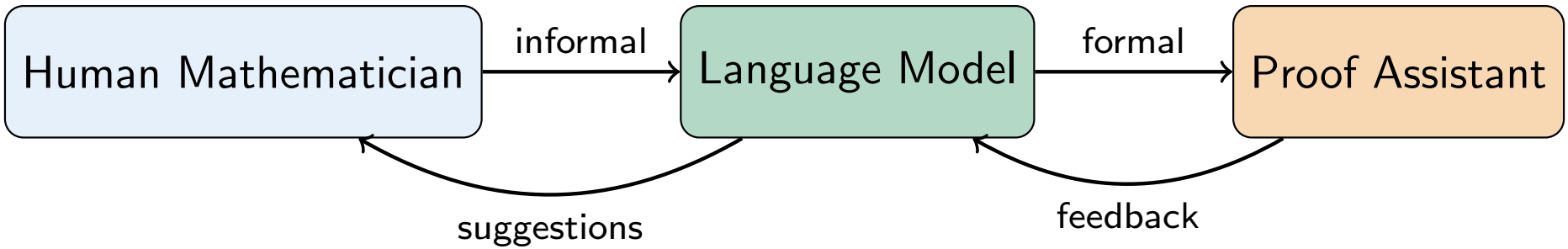
Modus Ponens

```
-- If P implies Q,
-- and P holds, then Q holds

theorem modus_ponens
  (P Q : Prop)
  (hpq : P -> Q)
  (hp : P) : Q := by
  apply hpq
  exact hp
```

De Morgan's Law

```
-- not (P and Q) iff
-- (not P) or (not Q)

theorem de_morgan
  (P Q : Prop) :
  not (P or Q) <-> (not P and not Q) :=
    by
  constructor
    intro h
    by_cases hp : P
      right; intro hq
      exact h (hp, hq)
      left; exact hp
    intro h (hp, hq)
    cases h with
    | inl hnp => exact hnp hp
    | inr hnq => exact hnq hq
```

AI + Proof Assistants: The Frontier



Recent breakthroughs:

- **AlphaProof** (DeepMind, 2024): Solved IMO problems using Lean
- **LeanDojo** (2023): AI agents that can prove Mathlib theorems
- **GPT-4 + Lean**: Autoformalization of natural language proofs

The Vision

AI translates human intuition into rigorous formal proofs, while proof assistants guarantee correctness.
[arxiv.org](https://arxiv.org/html/2603.03684v2)

AlphaProof at IMO 2024

In July 2024, DeepMind's **AlphaProof** system achieved:

- Solved 4 of 6 IMO problems
- Combined score: **Silver medal level**
- First AI to achieve this performance
- Used **Lean 4** for verification

How it works:

1. Language model proposes proof steps
2. Lean verifies each step
3. Reinforcement learning from success/failure
4. Self-play to generate training data

IMO Problem 1 (2024)

Determine all real numbers α such that for every positive integer n, the integer

$$\lfloor \alpha \rfloor + \lfloor 2\alpha \rfloor + \cdots + \lfloor n\alpha \rfloor$$

is divisible by n.

AlphaProof solved this in Lean.

Hands-On: Using AI for Mathematical Proofs

Exercise Framework

We will work through three types of exercises:

Level 1: Known Theorems

Use AI to:

- Explain proofs
- Find alternative approaches
- Translate to formal language

Level 2: Exercises

Use AI to:

- Get hints
- Verify steps
- Explore edge cases

Level 3: Open Problems

Use AI to:

- Explore structure
- Generate conjectures
- Find patterns

Important Reminder

AI can make mistakes! Always verify critical steps independently.

Exercise 1: Infinitude of Primes

Theorem (Euclid)

There are infinitely many prime numbers.

Task: Ask an AI assistant to:

1. Provide Euclid's original proof
2. Give an alternative proof (e.g., using $\sum 1/p$)
3. Sketch a formalization in Lean

Sample Prompt

"Prove that there are infinitely many primes. First give Euclid's proof, then show why $\sum_{p \text{ prime}} 1/p$ diverges, and finally show me how this would look in Lean."

Discussion: Compare the AI's response with the Mathlib formalization.

Exercise 2: Irrationality of $\sqrt{2}$

Theorem

$\sqrt{2}$ is irrational.

Task: Explore multiple proof strategies with AI:

1. Classic proof by contradiction (parity argument)
2. Geometric proof using isoceles right triangles
3. Proof using unique factorization

Classic Approach

Assume $\sqrt{2} = \frac{p}{q}$ in lowest terms.
Then $2q^2 = p^2$, so p is even.
Write $p = 2k$, then $q^2 = 2k^2$, so q is even.
Contradiction!

Prompt to try:
"Can you show me a geometric proof of the irrationality of $\sqrt{2}$ using an infinite descent argument with triangles?"

Exercise 3: Exploring Goldbach's Conjecture

Goldbach's Conjecture (1742)

Every even integer greater than 2 can be expressed as the sum of two primes.

Status: **Unproven** — but verified computationally for $n < 4 \times 10^{18}$

Task: Use AI to explore:

1. Write code to verify for small n
2. Investigate the number of representations $r(n)$
3. Discuss why this is hard to prove
4. Explore related conjectures (weak Goldbach, proved 2013)

Sample Investigation

"Write Python code to find all ways to write even numbers up to 100 as sums of two primes. Then plot the number of representations."

Exercise 4: The Collatz Conjecture

Collatz Conjecture (1937)

Define $f(n) = \begin{cases} n/2 & \text{if } n \text{ even} \\ 3n+1 & \text{if } n \text{ odd} \end{cases}$

Conjecture: For any positive integer n, repeated application of f eventually reaches 1.

Status: **Unproven** — Erdős said "Mathematics is not yet ready for such problems."

Task: Explore computationally with AI:

1. Generate sequences for various starting points
2. Analyze "stopping times" (steps to reach 1)
3. Visualize the "Collatz graph"
4. Discuss partial results (e.g., Tao 2019)

Tao's Result (2019)

Almost all Collatz orbits attain almost bounded values.

Exercise 5: Formalization Challenge

Challenge: Work with AI to formalize a simple theorem.

Target Theorem

For all natural numbers n: $\sum_{k=0}^{n} k = \frac{n(n+1)}{2}$

Steps:

1. Ask AI to write a Lean proof
2. Understand each tactic used
3. Try to modify for $\sum k^2 = \frac{n(n+1)(2n+1)}{6}$

Expected Lean Structure

```
theorem sum_range (n : Nat) :
  2 * (Finset.range (n+1)).sum id = n * (n + 1) := by
  induction n with
  | zero => simp
  | succ n ih =>
    rw [Finset.sum_range_succ]
    ring_nf
    linarith
```

Best Practices for AI-Assisted Mathematics

DO:

- ✓ Verify AI outputs independently
- ✓ Use AI for exploration and intuition
- ✓ Ask for multiple approaches
- ✓ Request explanations of steps
- ✓ Cross-check with formal tools
- ✓ Treat AI as a collaborator

DON'T:

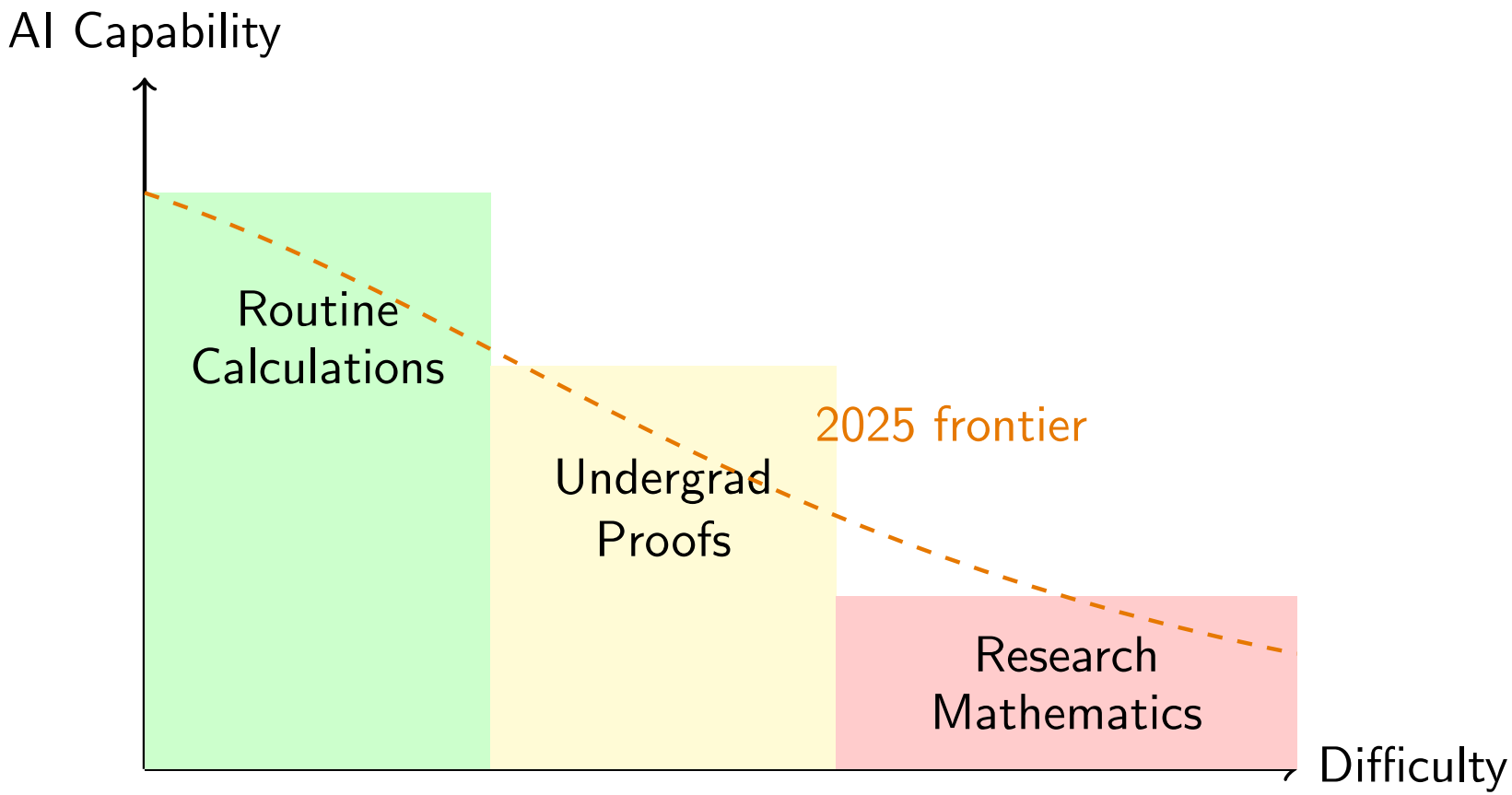
- × Trust calculations blindly
- × Accept "proofs" without checking
- × Assume AI understands deeply
- × Skip learning fundamentals
- × Use AI for exams dishonestly
- × Forget to cite AI assistance

Critical Thinking

AI can produce convincing-sounding but incorrect proofs. **Mathematical validity requires verification, not just plausibility.**

Conclusions and Future Directions

The State of AI in Mathematics (2025)



"Recent developments show that AI can prove research-level theorems in mathematics, both formally and informally." [arxiv.org](https://arxiv.org/html/2603.03684v2)

Looking Ahead: Open Questions

1. **Understanding vs. Pattern Matching**
   - Do LLMs "understand" mathematics or mimic patterns?
   - Can they generate genuinely novel proofs?
2. **Human-AI Collaboration**
   - How will mathematical research change?
   - What skills will future mathematicians need?
3. **Verification and Trust**
   - Can we trust AI-generated proofs?
   - Role of formal verification as the "ground truth"
4. **Creativity and Discovery**
   - Can AI discover new mathematics?
   - Will AI solve major open problems?

Resources for Further Exploration

Proof Assistants:

- Lean: leanprover.github.io
- Natural Number Game: adam.math.hhu.de
- Mathlib documentation: leanprover-community.github.io

Reading:

- Tao, T. (2024). "Machine Assisted Proofs" [terrytao.wordpress.com](https://terrytao.wordpress.com/wp-content/uploads/2024/03/machine-jan-3.pdf)
- Avigad, J. (2026). "Mathematicians in the Age of AI" [arxiv.org](https://arxiv.org/html/2603.03684v2)
- Herman, R.L. (2025). "AI in Mathematics Research" [people.uncw.edu](https://people.uncw.edu/hermanr/mat495/AI_in_Math.pd

Summary

Key Takeaways

1. **Mathematics powers AI**: Linear algebra, calculus, probability, and logic are the foundation
2. **AI assists mathematics**: Computer-verified proofs are increasingly important
3. **Historical landmarks**: Four Color Theorem, Kepler Conjecture, and more
4. **Proof assistants**: Lean, Coq, Isabelle provide formal verification
5. **The frontier**: AI can now prove IMO-level and some research-level theorems
6. **Critical thinking**: Always verify AI outputs — they can be confidently wrong

Questions?

Thank You

Mathematics + AI = The Future of Discovery

"The computer is incredibly fast, accurate, and stupid.
Man is incredibly slow, inaccurate, and brilliant.
The marriage of the two is a force beyond calculation."

— Often attributed to Leo Cherne