UNIVERSITY OF ILLINOIS SPRINGFIELD

AI and Mathematics

How should we learn, prove, and do mathematics now?

Xiang Huang
Department of Computer Science
University of Illinois Springfield

November 1, 2026



Outline

My research

Formalization

Celebrate, not fear

Common worries

Teaching

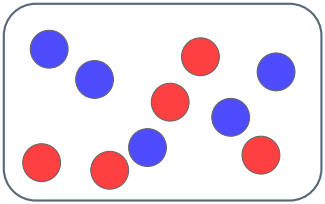
Correctness

Resources

What is a population protocol?

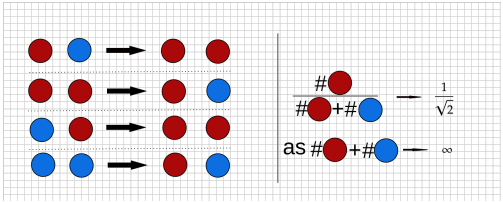
- A computer made of a pool of N balls, each with a color
- Pick two balls at random and apply a rule: **two in, two out**
- Repeat, again and again; no one is in charge
- The answer, in this talk: the **fraction** of balls of one color, in the long run (one way to read the answer out; there are many others)

Our job: design the rules, so that this fraction becomes the number we want.



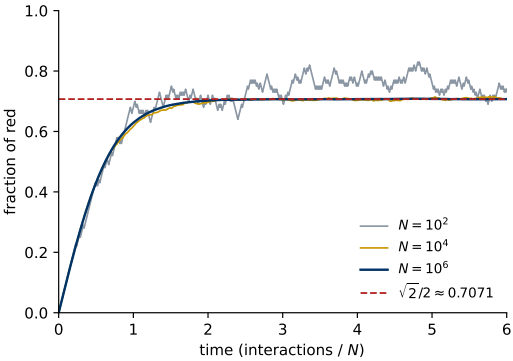
a rule: red meets blue, blue turns red

A population protocol can compute a number



Protocol of Aupy and Bournez (2011)

- N individuals, red or blue, meet in random ordered pairs
- Red fraction x follows $x' = 2x(1-x) + (1-x)^2 - x^2 = 1 - 2x^2$
- So $x \to \frac{1}{\sqrt{2}} = \frac{\sqrt{2}}{2}$ as $N \to \infty$



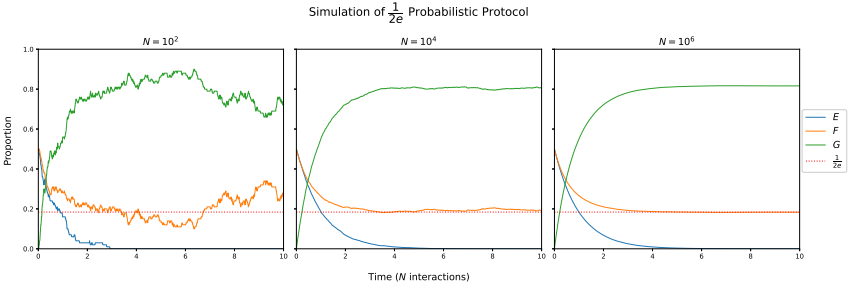
Even a transcendental number

Main theorem

Large-population-protocol computable numbers = numbers computable by polynomial ODEs (GPAC).

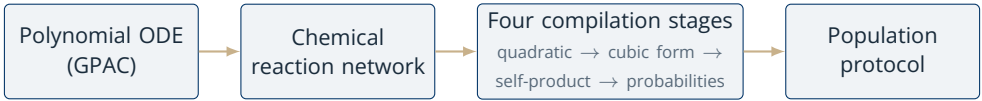
Example. Start with half F, half E:

$$F + E \xrightarrow{2} G + E, \quad E \to G \quad \Longrightarrow \quad \frac{\#F}{N} \to \frac{1}{2e}$$



A transcendental number from random pairwise interactions; larger N, sharper limit.

My papers: compiling ODEs into population protocols



Every step needs different mathematics

- Differential equations: solutions stay bounded and converge to the right limit
- Algebra: rewrite the polynomials into forms a population protocol can implement
- Probability: as $N \to \infty$ the random population follows the ODE (Kurtz's theorem)
- Number theory: which numbers are reached — $\pi/4$, Euler's γ, Catalan's constant

Huang–Klinge–Lathrop 2019; Huang–Huls, DNA 28 (2022); Chen–Huang, DNA 32 (2026)

A full example: Euler's constant $\gamma = 0.5772\ldots$

Start from a formula

$$\gamma = 1 - \Gamma'(2), \qquad \Gamma'(2) = e^{-1}(\alpha + \beta),$$
$$\alpha = \int_0^1 e^s (1-s)\log(1-s)\,ds, \qquad \beta = \int_0^\infty e^{-s}(1+s)\log(1+s)\,ds$$

- Goal: a population protocol whose output fraction converges to γ
- Route: formula $\to$ ODE $\to$ chemical reactions $\to$ population protocol

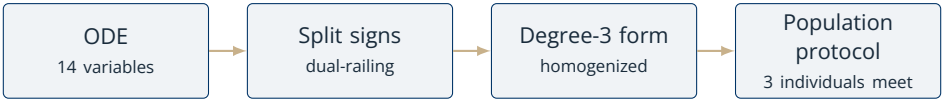
Construction from Huang's PhD thesis (2020), Thm. 3.3.4

Step 1: an ODE whose solution computes γ

$$
\begin{aligned}
f' &= w & f(0) &= 0 \\
g' &= -pqv & g(0) &= 0 \\
w' &= -w + u + v & w(0) &= 0 \\
u' &= -u + rv & u(0) &= 0 \\
v' &= -v & v(0) &= 1 \\
r' &= -r^2 & r(0) &= 1 \\
p' &= pv & p(0) &= 1 \\
q' &= v - q & q(0) &= 0
\end{aligned}
$$

- $v = e^{-t}$, $r = \frac{1}{1+t}$, $q = te^{-t}$, ...
- $f(t) \to \beta$ and $g(t) \to \alpha$ as $t \to \infty$
- A few more variables turn α, β into γ itself: **14 variables**
- Polynomial right-hand sides, rational initial values, bounded solutions

Step 2: compile

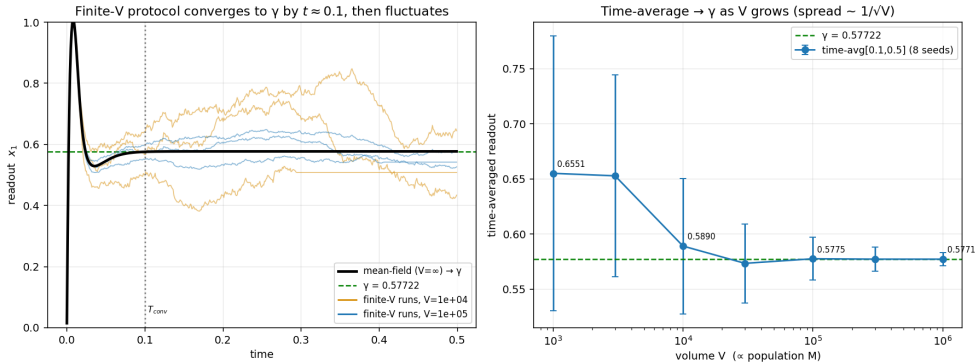


149
states

3420
interaction rules

Far too large to write on a slide; the compiler generates it automatically.

Step 3: simulate



Left: the ODE reaches γ by $t \approx 0.1$; finite populations fluctuate around it. Right: the time average approaches γ as the population grows.

Exact stochastic simulation with $\sim 2 \times 10^8$ molecules: 0.577 ± 0.004.

Two worries

01

Is it correct?
I cannot be sure my papers contain no errors.

02

Can I master every field it needs?
Chemical reaction networks, ODEs, number theory, transcendental numbers, stochastic analysis, measure theory — mastering all of them in depth takes a long time.

UNIVERSITY OF ILLINOIS SPRINGFIELD

SECTION 2

Then AI arrived

An answer to both worries: machine-checked proofs, and a collaborator who reads every field.



Formalization is an old dream

Long before machines

- Euclid (c. 300 BC): axioms and deductions
- Leibniz (1646–1716): a calculus of reasoning
- Dedekind (1831–1916): numbers from first principles
- Hilbert (1862–1943): all of mathematics as a formal system

Theory comes first

Formal systems before proof assistants.
Algorithms before computers.
The idea of AI before AI.

Lean: mathematics a computer can check

Mathematics (Peano)

Only two things:

- 0 is a number
- if x is a number, so is $\operatorname{succ} x$

So the numbers are
$0,\ \operatorname{succ} 0,\ \operatorname{succ}\operatorname{succ} 0,\ \ldots$

Addition, by two rules:

1. $0 + y = y$
2. $\operatorname{succ} x + y = \operatorname{succ}(x + y)$

Lean

```
inductive N where
  | zero : N          -- 0
  | succ : N -> N     -- succ x
open N

def add : N -> N -> N
  | zero,   y => y                -- rule 1
  | succ x, y => succ (add x y)   -- rule 2
```

Proving $x + 0 = x$

On paper

Not one of the rules, so we must prove it. Induction on x.

$x = 0$**:** $0 + 0 = 0$ by rule 1.

$x \to \operatorname{succ} x$**:** assume $x + 0 = x$. Then

$$\begin{aligned} \operatorname{succ} x + 0 &= \operatorname{succ}(x + 0) && \text{rule 2} \\ &= \operatorname{succ} x && \text{assumption} \end{aligned}$$

In Lean

```
theorem add_zero (x : N) :
    add x zero = x := by
  induction x with
  | zero =>          -- 0 + 0 = 0
    rw [add]         -- rule 1
  | succ x ih =>     -- ih: x + 0 = x
    rw [add]         -- rule 2
    rw [ih]          -- assumption
```

Lean checks every step; a gap is an error.

Then, late 2025 to early 2026

Lean itself is not intelligent: a logic and a programming language, with some clever tactics. But large language models learned to write it. How good are they? I started trying.

My formalization work since spring 2026

- **Ripple**: computing with chemical reaction networks, my own research and related papers in the field (about 800k lines)
- **A PDE trilogy** of Wenxian Shen et al. on chemotaxis (about 800k lines)
- **Recurrence limits** for classical constants (π, γ, $\zeta(3)$, ...)
- ***Proofs from THE BOOK***: all 40 chapters
- ***An Invitation to q-Series***: the whole book (255k lines)
- ***Analytic Combinatorics***: all nine chapters

xianghuang.org/mypages/ai-math.md

2.5M

lines of machine-checked Lean 4

AI writes. Lean checks.

About four to five months with AI.
Without AI: years of work.

How a proof gets in

The workflow

1. AI drafts Lean from the paper proof
2. Lean checks every step
3. errors go back to the AI to repair

Three gates

1. the statement is the theorem we intended
2. no placeholders, no custom axioms
3. a fresh build passes

Errors found

Three projects turned up errors in published proofs.

In Ripple, for example: a compilation step that can briefly leave the unit interval, and an algebraic-number construction that silently relied on Catalan's conjecture.

How far can AI formalization go? Fermat's Last Theorem

- **May 2026.** Kevin Buzzard, leading the human FLT project (5 years, £1M), plans a workshop: "a challenge" to those who think autoformalization is solved
- **August 2026.** Claude, largely autonomously, in 11 days: the first complete computer-checked proof of FLT, using only Lean's standard axioms
- **September 2026.** Buzzard: "Anthropic has beaten me to it."

13M lines of Lean
29,500 theorems
11 days

Buzzard

"If thousands of pages of the literature can be formalized end-to-end [...] in an 11 day period now, then in the future we will start to see formalization of modern research being done on the fly."

Two irrationality proofs, September 2026

$\zeta(5)$ is irrational

Fauzan, Sept. 17
Complete Lean proof, no gaps.

Catalan's constant is irrational

Sun, Sept. 3
"The whole proof has passed the verification of Chatgpt 5.6 Solar". No formal proof.
Sept. 16: a gap is found in the asymptotic estimates.

Talking with a chatbot is not verification.

A formal proof settles the question.

And with AI, writing one is now within everyone's reach.

AI reads every field

- Mastering every branch in depth takes years; AI can consult all of them at once
- Crossing fields used to need many collaborators
- Final confidence may still need people — for now

Ramanujan, 1913

A clerk at the Madras Port Trust with no university education. Local mathematicians called his results "startling" but could not judge them, so he wrote to Hardy in Cambridge for help.

Today, anyone without such resources can have a collaborator with broad mathematical knowledge: AI.

Mathematics has its AlphaGo moment

- 2016: AlphaGo beats Lee Sedol, 4–1
- 2017: AlphaGo beats Ke Jie, 3–0
- A lifetime of work, overtaken by a machine
- Then players studied the AI's games: new moves learned, old ones dropped

Solved with AI, 2026

Jan	Erdős Problem #728, on factorials	GPT-5.2 Pro and Aristotle; checked in Lean
Feb	Knuth's Hamiltonian-cycle problem from *The Art of Computer Programming* (odd m)	Claude Opus 4.6: "Claude's Cycles"
May	Erdős's unit distance conjecture (1946) disproved, with algebraic number theory	an OpenAI model
Aug	A Hadamard matrix of order 668, the smallest open case for twenty years	Alpöge, Voinov, Reynolds-Haertle, and Claude
Sept	Blow-up in three dimensions: Euler, then Navier–Stokes (a Millennium Problem); checked in Lean	Buckmaster and Alpöge (Euler), OpenAI (Navier–Stokes), both with AI, building on Córdoba and Martínez-Zoroa

A short list, far from complete: the examples keep coming.

We must know, we will know

Hilbert, Königsberg, 1930

"We must not believe those who today, with philosophical bearing and superior tone, prophesy the fall of culture … Our slogan shall be: We must know, we will know."

- Hilbert: the goal is to know; do not be stopped by prophets of decline
- Today's prophecy of decline: "AI will ruin mathematics"
- But if AI helps us know, the goal comes closer. An open problem answered is curiosity winning: celebrate it

The Foolish Old Man who moved the mountains

- A Chinese fable from the *Liezi*
- An old man, nearly ninety, sets out to move two great mountains that block his door
- His family chisels the rock and carries it away in baskets
- The story is a hymn to human courage



Mathematics is our mountain: what we want to know.

Moving mountains



Chisels and baskets

Mathematicians with pen and paper



Excavators and blasting

Mathematicians with AI and Lean, today



Driverless excavators

Autonomous AI, tomorrow

Will there be nothing left to do?

After one mountain there is a bigger one.

Mathematics: from heroes to engineers

Noah Smith, *The end of the age of heroes* (Aug. 2026)

"Every idea that got added to the collective ultimately had to be comprehensible by a single human being."

- That bottleneck made heroes: Newton, Wiles, Perelman
- AI removes it: 13M lines of Lean fit in no single head, yet every line is checked
- Mathematics becomes a large engineering project: teams, machines, verification
- For everyone who uses mathematics as a tool, this is a golden age

Big projects need big tools

- Is mathematics now an arms race for compute? A big project does need big resources and organization
- No one builds a dam, or moves a mountain, with a chisel
- We need new tools and new engineering methods

10,000

AI agents for 88 hours:
OpenAI's Navier–Stokes proof
(Sept. 2026)

My view

Asking people to keep chiseling, one stroke at a time, is not realistic.

Where do we fit in?

- Costly does not mean closed. Choose well: contribute one part, as collaborators always have, or pick a smaller project
- People have their role, machines have theirs; together we will find a good way to work

Example: Prove2me (prove2.me)**, open theorems proved in Lean by people with AI agents**

The human's part: write the formal statements, lay out and track the milestones, and work with your agents. **The more you know, the more you can do.**

“Too much to check”

- Major results now ship with Lean code: we know what is correct
- Machine output will not pollute our knowledge, if it is checked in Lean
- What does not work: a chatbot’s approval (recall Catalan’s constant)

“Too much to understand”

Understood by whom?

- A formal proof joins human knowledge the moment it exists
- It can be mined, reused, refined to any level of detail
- Anyone can study it, in parallel
- It is finite: 13M lines can be broken down and digested

Go players did not say “we cannot understand it”

They learned from it. No one has to understand everything: follow what your own questions need.

“Nothing will be left for us”

- **Low-hanging fruit?** Yesterday’s is gone in one scoop of the excavator; we redefine it: the excavator era has its own
- **Basic training?** It is basic mathematics, not open problems, and it barely changes: teach elementary school and undergraduate mathematics as before
- **Fewer mathematicians?** Combine harvesters meant fewer farmers, enough food, and people free for new work. Mathematical skill and knowledge still matter: they are what let you use the machines well
- **Credit and evaluation?** Review is too slow to keep up: the system must change
  - Much review effort goes into checking correctness
  - Submit with a Lean formalization: correctness is settled at submission
  - Reviewers then judge the presentation and the explanation

“AI takes the struggle out of research”

37

rounds with AI before I found my construction

- Try an idea, understand why it fails, propose a new one, fail again
- That loop has not changed; the struggle is still there
- What changed: AI does the intermediate checking and calculation, fast and well
- So each round is faster: the iteration speeds up

Models used then: Opus 4.6 and GPT-5.5. Today’s models are smarter, but the struggle depends on the problem: if AI finds it easy, the problem was already close to solved and needed no deep search.

What should we teach?

Foundations

- Closed-AI, at least for the basics
- The memory burden is worth bearing
- Students need their own understanding

AI tools

- In my class, almost no one had used Claude or Codex
- They are production tools for the workforce
- Learn by doing, not a dedicated course: the tools change too fast

A question for all of us, not only students

With AI this strong, what rules should research follow, so that we still learn from it? We are writing these rules now, together.

Back to “Is it correct?”

01 **Being correct matters, whether or not we understand it.**
A correct result is useful before anyone understands it in depth.

02 **Example: verification.**
Programs, security protocols, hardware designs can all be modeled mathematically; our work and lives depend on their being correct. AI finds Linux kernel bugs faster; better still is correct by design.

03 **Nothing here is incomprehensible.**
Every line of the proof is there, to read, check and reuse.

Where to start

01 **Large language models.**
Andrej Karpathy, *Neural Networks: Zero to Hero*
karpathy.ai/zero-to-hero.html

02 **Lean.**
The *Natural Number Game*: prove $a + b = b + a$ from scratch
adam.math.hhu.de
For general readers: Kevin Hartnett, *The Proof in the Code* (2026)

03 **AI agents.**
Learn by doing: build something small on GitHub with a coding agent (Claude Code, Codex, GitHub Copilot)

04 **Open models.**
Qwen, DeepSeek, Llama: low-cost models for learning the basics

UNIVERSITY OF ILLINOIS SPRINGFIELD

Questions?

Xiang Huang
Department of Computer Science
University of Illinois Springfield