Unpacking the Navier–Stokes blowup: the mathematics, the machines, and the mathematicians

Kolen Cheung

September 11th, 2026

Abstract

OpenAI claims a solution to alternatives C/D of the Navier–Stokes Millennium Prize Problem. In the same week, two other groups announced finite-time blowup results for Euler using different approaches. This post is trying to unpack these results for someone with undergraduate mathematics and physics and some knowledge of LLMs and agentic engineering.

We start from the physics, write down the four Clay statements, and go through the idea of the construction. A collapsing vortex spins up by conservation of angular momentum, the same reason water turns faster as it nears a drain. High-frequency corrections provide the averaged momentum flux needed to keep the force smooth. This also explains how the speed can diverge while the total energy stays bounded, why Navier–Stokes blowup requires unbounded speed, and how rescaling gives the construction for every positive viscosity.

Then we discuss Lean, which checked two of the three results before human review. A proof can be correct and still be hard to learn from, as illustrated by an author calling his own Lean-verified paper "AI slop". We still need to check that the formal statement says what we intend and that the proof uses only the allowed assumptions. On the ethics, I think OpenAI probably did not take anyone's work; the different approaches support that. But two groups rushed unfinished work out, and Tao worries that open problems are being mined non-renewably. I think this is related to how mathematics assigns credit, from Newton–Leibniz to Perelman to the reception of computer-assisted proofs. Could open development help? We also discuss what happens when proofs cost millions of dollars, and some results may be worth keeping secret. Finally, we look at the role of rumours, scale, and verifiable targets, why I doubt mathematics will have an AlphaGo moment, and what the mathematical community and AI labs could do differently.

Table of contents

1 Introduction

There's a lot of buzz right now about a claimed proof of "Breakdown of Navier–Stokes solutions". It is one of the Millennium Prize Problems, and a solution to its breakdown alternatives is claimed by OpenAI (OpenAI 2026c).

This post is trying to unpack this for a "layperson", by which I mean someone with undergraduate mathematics and physics and some knowledge of computer science, recent LLMs, and agentic engineering.

First, what are the Millennium Prize Problems?

> The Prizes were conceived to record some of the most difficult problems with which mathematicians were grappling at the turn of the second millennium; to elevate in the consciousness of the general public the fact that in mathematics, the frontier is still open and abounds in important unsolved problems; to emphasize the importance of working towards a solution of the deepest, most difficult problems; and to recognize achievement in mathematics of historical magnitude (Clay Mathematics Institute, n.d.).

There are seven of them, and if you click the link and take a look at the list, I'd argue that the Navier–Stokes Equation prize is the one requiring the least background to understand the content, the statement, and the *idea* behind the construction.[1] So we are in for a treat—we have a math problem so important that it deserves a Millennium Prize, yet with an idea we can unpack at an undergraduate level.

As for the prize itself, Fefferman's official description offers four statements, A–D, and says he is stating them this way "to give reasonable leeway to solvers while retaining the heart of the problem" (Fefferman 2000). A proof of C therefore counts. The Clay rules also require publication in a qualifying outlet, at least two years since publication, and general acceptance by the mathematical community (Clay Mathematics Institute 2018). OpenAI has said it does not intend to claim the prize anyway (OpenAI 2026c).

2 Motivation from physics

The underlying physics is very simple. We assume

- Newtonian (classical) dynamics, in the continuum limit;
- a *Newtonian fluid*: the viscous stress is proportional to the rate of strain, with the constant of proportionality (the viscosity) the same everywhere;
- an incompressible fluid of uniform density;
- ~~common sense~~ conservation of mass.

Then we'll arrive at the Navier–Stokes equations—simple equations with just enough physics to be interesting—

$$\boxed{\begin{aligned}\partial_t u + (u\cdot\nabla)u &= -\nabla p + \nu\Delta u + f,\\ \nabla\cdot u &= 0.\end{aligned}} \tag{1}$$

$u = \vec{u}(\vec{x},t)$ is the velocity field[2] of the fluid.

$f = \vec{f}(\vec{x},t)$ is the external(ly provided) force field—an acceleration, like g.

$p = p(\vec{x},t)$ is the pressure field, and $\nu > 0$ is the viscosity. p has an extra gauge degree of freedom: adding a function of time alone has no effect on equation 1.[3]

We have divided out the density in defining p, ν, f, effectively setting it to one. To physicists, think of conventions like $c = \hbar = 1$, or Gaussian units in EM. The physical derivation in section 2.1 shows how this works. For the mathematical problem we can use these rescaled quantities without carrying density around.

[1] Let me put it this way: If there exists a proof of the Riemann Hypothesis, I would not be able to unpack it, probably.

[2] In physics, we call something a field if it depends on space and time. Hence, this is classical field theory.

[3] Choosing the reference is called fixing the **pressure gauge**. On $\mathbb{T}^3$, a common choice is $\int_{\mathbb{T}^3} p(x,t)\,dx = 0$ at every time. On $\mathbb{R}^3$, with suitable decay, use $p(x,t) \to 0$ as $|x| \to \infty$.

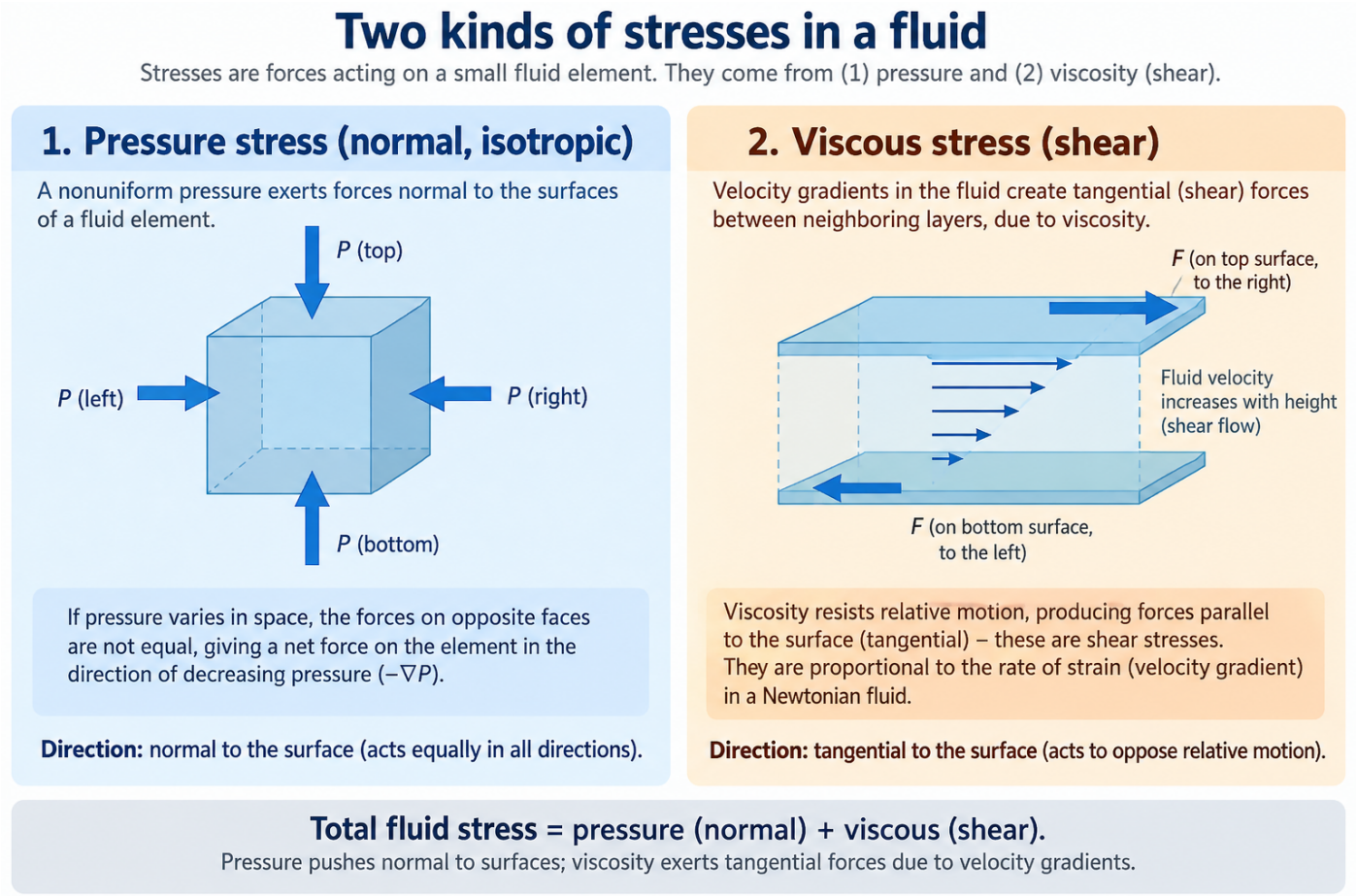

Figure 1: Pressure and viscous stress on a fluid element. Illustration generated by the author using OpenAI Images 2.5; schematic, not a simulation.

Given f, ν, if we specify an initial condition $u(x,0) = u_0(x)$[4], we expect to find a solution (u,p).[5]

2.1 Digression: physical derivation

(Skip this section if you already know the physics, or don't care.)

Starting from the classical (Newtonian) particle perspective,

$$\sum_i m_i \mathbf{a}_i = \sum_i \mathbf{F}_i^{\text{ext}} + \sum_{i,j} \mathbf{F}_{ij}. \tag{3}$$

On the right of equation 3, the first term is external force and the second is internal force. Internal forces cancel when summed over the whole system; for a small fluid element, the surrounding fluid exerts stress across its boundary.

To transition to the continuum limit, $dm = \rho\, dV$.

Stress has two contributions, illustrated in figure 1. Strictly, stress is force per unit area. The viscous panel shows shear; viscous stress can also have normal components, as equation 4 allows. A difference in physical pressure P introduces this term: $d\mathbf{F}_P = -\nabla P\, dV$. Viscous stress (including shear) is what makes it a "Newtonian fluid": the stress is linear in the rate of strain,

$$\tau = \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^\top), \tag{4}$$

[4]which must also satisfy the divergence-free condition

[5]How is p obtained? The condition $\nabla \cdot u = 0$ alone does not determine pressure. Requiring the momentum equation to **preserve** this condition does: taking its divergence, with $\nabla \cdot \partial_t u = 0$ and $\nabla \cdot \Delta u = 0$, gives

$$-\Delta p = \sum_{i,j=1}^{3} (\partial_i u_j)(\partial_j u_i) - \nabla \cdot f. \tag{2}$$

To solve this Poisson equation we also need spatial conditions: periodicity, or suitable behaviour at infinity. The remaining function of time is fixed by the pressure gauge.

and its divergence is the viscous force per unit volume. With constant μ and $\nabla \cdot \mathbf{u} = 0$ the second term drops out and this collapses to $\nabla \cdot \tau = \mu \nabla^2 \mathbf{u}$, giving $d\mathbf{F}_{\text{visc}} = \mu \nabla^2 \mathbf{u}\, dV$.

Here $\mathbf{u}$ is velocity and $\mathbf{f}$ is external force **per unit mass** (e.g. gravitational acceleration).

Putting everything together, for a fluid element of volume dV:

$$\rho\, dV \frac{D\mathbf{u}}{Dt} = -\nabla P\, dV + \mu \nabla^2 \mathbf{u}\, dV + \rho \mathbf{f}\, dV. \tag{5}$$

In equation 5, the left-hand side is ma; the three terms on the right are pressure, viscous, and external forces.

Expanding the material derivative (following a moving fluid element)

$$\frac{D\mathbf{u}}{Dt} = \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u}. \tag{6}$$

Combining the assumption of constant density ρ and conservation of mass: $\partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0$.[6]

We have

$$\boxed{\nabla \cdot \mathbf{u} = 0.} \tag{7}$$

Combining equations 5 and 6 gives

$$\boxed{\rho \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} \right] = -\nabla P + \mu \nabla^2 \mathbf{u} + \rho \mathbf{f}.} \tag{8}$$

Divide by ρ and define the kinematic pressure and viscosity:

$$p := \frac{P}{\rho}, \qquad \nu := \frac{\mu}{\rho}. \tag{9}$$

Using equation 9, write the vectors $\mathbf{u}, \mathbf{f}$ as u, f, and ∇^2 as Δ:

$$\boxed{\partial_t u + (u \cdot \nabla)u = -\nabla p + \nu \Delta u + f.} \tag{10}$$

- $(u \cdot \nabla)u$: the fluid carries its own velocity field along — the nonlinear term.
- $\nu \Delta u$: viscosity smooths out velocity differences, like diffusion.
- $-\nabla p$: pressure adjusts to keep $\nabla \cdot u = 0$.

3 Mathematical statement of the Millennium Prize Problem

$$\begin{aligned} u &: \Omega \times [0, T) \to \mathbb{R}^3, \\ p &: \Omega \times [0, T) \to \mathbb{R}, \\ f &: \Omega \times [0, \infty) \to \mathbb{R}^3, \\ u_0 &: \Omega \to \mathbb{R}^3, \quad u_0(x) = u(x, 0). \end{aligned} \tag{11}$$

$$\boxed{\begin{aligned} \partial_t u + (u \cdot \nabla)u &= -\nabla p + \nu \Delta u + f, \\ \nabla \cdot u &= 0, \\ u(x, 0) &= u_0(x). \end{aligned}} \tag{12}$$

T: time up to which the solution exists; global means $T = \infty$.

Smooth means C^∞: continuous derivatives of every order, in both space and time where applicable, including one-sided limits at $t = 0$.

[6] Incompressibility and uniform density are separate assumptions. Incompressibility says a moving parcel preserves its volume, so with mass conservation its density stays constant along its motion. Different parcels could still have different densities. Here we also assume ρ is uniform, which lets us divide it out in equation 9.

3.1 The four Clay statements

There are four variants of the Millennium Prize statement, labelled A–D. The definitions follow Fefferman's official problem description and errata (Fefferman 2000, 1–2, 6).

3.1.1 The domain

$$\Omega \in \{\mathbb{R}^3,\ \mathbb{T}^3 = \mathbb{R}^3/\mathbb{Z}^3\}. \tag{13}$$

$\mathbb{R}^3$: all space.

$\mathbb{T}^3$: a unit cube with opposite faces identified (to physicists, think periodic boundary conditions); u_0, f, u, p are periodic in each spatial coordinate.

The torus is easier to follow because we do not need conditions on what happens far away. For simplicity, you can follow just the torus case below (variant D for blowup).

3.1.2 Admissible data

Admissible initial velocities are smooth and divergence-free,

$$\begin{aligned} \mathscr{U}_\Omega &:= \{u_0 \in X_\Omega : \nabla \cdot u_0 = 0\}, \\ X_\Omega &:= \begin{cases} \mathscr{S}(\mathbb{R}^3;\mathbb{R}^3), & \Omega = \mathbb{R}^3, \\ C^\infty(\mathbb{T}^3;\mathbb{R}^3), & \Omega = \mathbb{T}^3, \end{cases} \end{aligned} \tag{14}$$

and admissible forces $\mathscr{F}_\Omega$ are smooth on $\Omega \times [0,\infty)$. On $\mathbb{R}^3$, Clay also requires u_0 and f to decay rapidly in space. On both domains, f must decay rapidly in time.

One way to satisfy the decay conditions is to use data with **compact support**:

$$u_0 \in C_c^\infty(\Omega;\mathbb{R}^3), \qquad f \in C_c^\infty(\Omega \times [0,\infty);\mathbb{R}^3). \tag{15}$$

I.e. u_0 and f are zero outside a bounded region of space, and f stays zero after some finite time.[7] OpenAI's C/D construction has these properties; see section 3.2.

Clay only requires **rapid decay**: the functions and all their derivatives fall off faster than any inverse power.[8] Compact support is stronger than this, since the functions are zero beyond some point. Rapidly decaying functions can still have tails extending to infinity, like a localized wave function in QM.[9]

3.1.3 The four statements

Write $\mathsf{G}_{\Omega,\nu}(u_0,f)$ for: there exists a global smooth pair (u,p) solving equation 12, with the additional whole-space condition

$$\sup_{t\ge 0} \int_{\mathbb{R}^3} |u(x,t)|^2\,dx < \infty. \tag{16}$$

For each fixed $\nu > 0$:

[7] The support is the closure of the set where the function is nonzero; compact means closed and bounded in Euclidean space. On $\mathbb{T}^3$, space is already compact, so for u_0 the condition is empty and for f only the behaviour in t is a restriction. Smoothness must hold through any proposed blowup time; smooth only for $t < T$ is insufficient.

[8] Precisely, for every multi-index α, every integer $m \ge 0$, and every integer $N \ge 0$,

$$|\partial_x^\alpha u_0(x)| \le C_{\alpha N}(1+|x|)^{-N} \quad (\Omega = \mathbb{R}^3),$$

$$|\partial_x^\alpha \partial_t^m f(x,t)| \le C_{\alpha m N} \begin{cases} (1+|x|+t)^{-N}, & \Omega = \mathbb{R}^3, \\ (1+t)^{-N}, & \Omega = \mathbb{T}^3, \end{cases}$$

with each C finite and independent of x, t, and $\partial_x^\alpha = \partial_{x_1}^{\alpha_1}\partial_{x_2}^{\alpha_2}\partial_{x_3}^{\alpha_3}$. The first line is what $\mathscr{S}(\mathbb{R}^3;\mathbb{R}^3)$, the Schwartz class, means in equation 14. On $\mathbb{T}^3$ there is no x to decay in, which is why $X_{\mathbb{T}^3}$ is just C^∞.

[9] C/D ask for one example, so compactly supported data suffice. A/B concern every admissible u_0; proving A only for compactly supported data would not settle it. Rapid decay is stronger than square-integrability, but does not require exponential decay.

Table 1: The four Clay alternatives, for each fixed $\nu > 0$.

Variant	Domain Ω	Statement
A	$\mathbb{R}^3$	$\forall u_0 \in \mathscr{U}_\Omega, \quad \mathsf{G}_{\Omega,\nu}(u_0, 0)$
B	$\mathbb{T}^3$	$\forall u_0 \in \mathscr{U}_\Omega, \quad \mathsf{G}_{\Omega,\nu}(u_0, 0)$
C	$\mathbb{R}^3$	$\exists u_0 \in \mathscr{U}_\Omega, f \in \mathscr{F}_\Omega : \quad \neg\mathsf{G}_{\Omega,\nu}(u_0, f)$
D	$\mathbb{T}^3$	$\exists u_0 \in \mathscr{U}_\Omega, f \in \mathscr{F}_\Omega : \quad \neg\mathsf{G}_{\Omega,\nu}(u_0, f)$

A/B: every admissible initial flow stays smooth without forcing. C/D: some admissible data admit no global smooth solution in the required class. Forcing is allowed in C/D, including $f = 0$; they are not the literal negations of A/B.

We fix $\nu > 0$, but can rescale the construction to work at any other positive viscosity; see section 3.3.5.

3.2 TL;DR and the 4 corners

In short, given the Navier–Stokes equations, you can specify u_0, f, the initial velocity and external force. Here we are trying to give both some nice properties: smooth (infinitely differentiable), and, on $\mathbb{R}^3$, rapidly decaying.

In OpenAI's construction, the fluid starts from rest, $u_0 = 0$, and the force has compact support in space and time. These are stronger conditions than required. The construction can also be shrunk to fit inside a cell of the torus and repeated periodically, so the same proof works for both C and D (OpenAI 2026a).

To reiterate: given nice properties of the input, is the solution to the Navier–Stokes equations guaranteed to stay nice for all time, or can it blow up (develop a singularity)? A physicist might expect there will never be a singularity, but it wouldn't be the first time a physicist is surprised. We should also distinguish the fluid from the model describing it. If the model predicts a singularity, perhaps one of its assumptions stops being valid before we get there? We'll come back to this in section 3.4.

The $\Omega = \mathbb{R}^3$ vs. $\mathbb{T}^3$ distinction is not central to the discussion today. However, there are two axes that are useful to know:

$$\nu > 0 \quad \text{Navier–Stokes}, \qquad \nu = 0 \quad \text{Euler},$$

$$f \neq 0 \quad \text{forced}, \qquad f = 0 \quad \text{unforced}.$$

Table 2: The four corners: viscosity and external forcing. Claims as of 11 September 2026.

	$f = 0$	$f \neq 0$
$\nu = 0$	unforced Euler: finite time blowup claimed by OpenAI; a separate self-similar candidate from Anandkumar's group	forced Euler: finite time blowup claimed by Alpöge/Buckmaster
$\nu > 0$	unforced Navier–Stokes (A/B): remains open	forced Navier–Stokes (C/D): finite time blowup claimed by OpenAI

The claims in table 2 come from OpenAI's two preprints, Alpöge and Buckmaster's announcement, and Anandkumar's account (OpenAI 2026a, 2026b; Buckmaster 2026b; Anandkumar 2026).

Note that Euler is not itself a Clay prize problem—Fefferman flags it as "also open and very important", but off the list (Fefferman 2000).

On 7 September, a Caltech group, Anima Anandkumar with Adash Ganeshram and Valentin Duruisseaux, with feedback from Tom Hou and Terence Tao, posted a self-similar blowup profile for unforced Euler on $\mathbb{R}^3$. They found it using physics-informed neural networks (PINNs) and used interval arithmetic to certify bounds on the approximate profile. LLMs assisted the stability analysis, which was partly formalized in Lean (Anandkumar 2026). This is still a candidate with computer-assisted bounds; the stability argument is not yet a finished proof. It is also independent of OpenAI's construction. As Anandkumar puts it, "their initial solution has a specific construction with a multi-scale structure with discrete jumps in scales. On the other hand, we consider a self-similar ansatz." (Anandkumar 2026) The timing will come up again in section 4.2.

3.3 Proofs

The claims in table 2 involve two approaches:

- Iterative corrections, from Córdoba–Martínez-Zoroa to Alpöge–Buckmaster and OpenAI: build up a singularity at increasingly fine scales, while keeping the initial data and, where applicable, the force smooth. High frequency here means short spatial wavelength.
- A self-similar profile, studied by Hou and collaborators over many years and now by Anandkumar's group: choose a profile that reproduces itself under rescaling, find it numerically, then certify it.

Both approaches try to construct a blowup, but so far only the first has produced claimed complete proofs for these equations. We will discuss that approach in section 3.3.3. Applying the idea to each equation requires further work.

Quoting Tao (2026b), on the forced Euler and related results of Alpöge and Buckmaster,

> The basic strategy, due to Cordoba and Martínez-Zoroa, is to iteratively build up the solution to such equations in stages, repeatedly adding small high frequency corrections to a previous (forced) solution in a manner that makes the solution more singular towards the blowup time while keeping the forcing term well behaved.

From here on we focus on OpenAI's forced Navier–Stokes construction, alternatives C/D. The iterative strategy described by Tao applies, but the corrections have a specific role: they provide the momentum transport needed to sustain a collapsing background flow. Without them, sustaining that flow would require a singular force (OpenAI 2026a).

3.3.1 What blows up?

Write the proposed blowup time as $T_* = 1$. The total kinetic energy (KE) stays bounded throughout:

$$E(t) = \frac{1}{2}\int_\Omega |u(x,t)|^2\,dx = \frac{1}{2}\|u(t)\|^2_{L^2(\Omega)}, \qquad \sup_{0\le t<1} E(t) < \infty. \tag{17}$$

And yet the peak speed becomes unbounded near $t = 1$:

$$\boxed{\limsup_{t\uparrow 1} \|u(t)\|_{L^\infty(\Omega)} = \infty.} \tag{18}$$

For Navier–Stokes, finite-time blowup requires the velocity to become unbounded near the blowup time T. This is a theorem, so any construction of a blowup must have this property (Fefferman 2000). This explains the use of $\|u\|_{L^\infty}$ in equation 18.

- For smooth u, $\|u(t)\|_{L^\infty} = \sup_{x\in\Omega} |u(x,t)|$; this alone does not say velocity diverges at one fixed point.
- Finite total energy can coexist with unbounded peaks concentrated in shrinking regions. In this construction the shrinking region's kinetic energy actually goes to *zero*, even as the speeds inside it diverge. We will see how in section 3.3.2.

For Euler, loss of smoothness need not mean unbounded speed. OpenAI's Euler theorem instead states

$$\limsup_{t\uparrow T_*} \|\nabla u(t)\|_{L^\infty} = \infty, \qquad \int_0^{T_*} \|\operatorname{curl} u(t)\|_{L^\infty}\,dt = \infty. \tag{19}$$

The vorticity condition in equation 19 comes from the **Beale–Kato–Majda** criterion: a smooth 3D Euler solution can continue past T if $\int_0^T \|\operatorname{curl} u\|_{L^\infty} dt < \infty$. Hence this integral must diverge for finite-time breakdown. It plays a similar role to the unbounded velocity condition for Navier–Stokes (OpenAI 2026b, Theorem 1.1; Fefferman 2000, 3).

Note that vorticity $\omega = \nabla \times u$ is a field measuring local rotation, while a vortex is a structure in the flow. We will describe the particular vortex in this construction in section 3.3.2.

How about energy conservation? Here the force can do work on the fluid, and viscosity dissipates kinetic energy. Dotting the momentum equation in equation 1 with u and integrating, using $\nabla \cdot u = 0$ and the spatial conditions, gives for $t < 1$

$$\frac{d}{dt}\left(\frac{1}{2}\|u(t)\|_2^2\right) + \nu\|\nabla u(t)\|_2^2 = \int_\Omega f \cdot u\,dx. \tag{20}$$

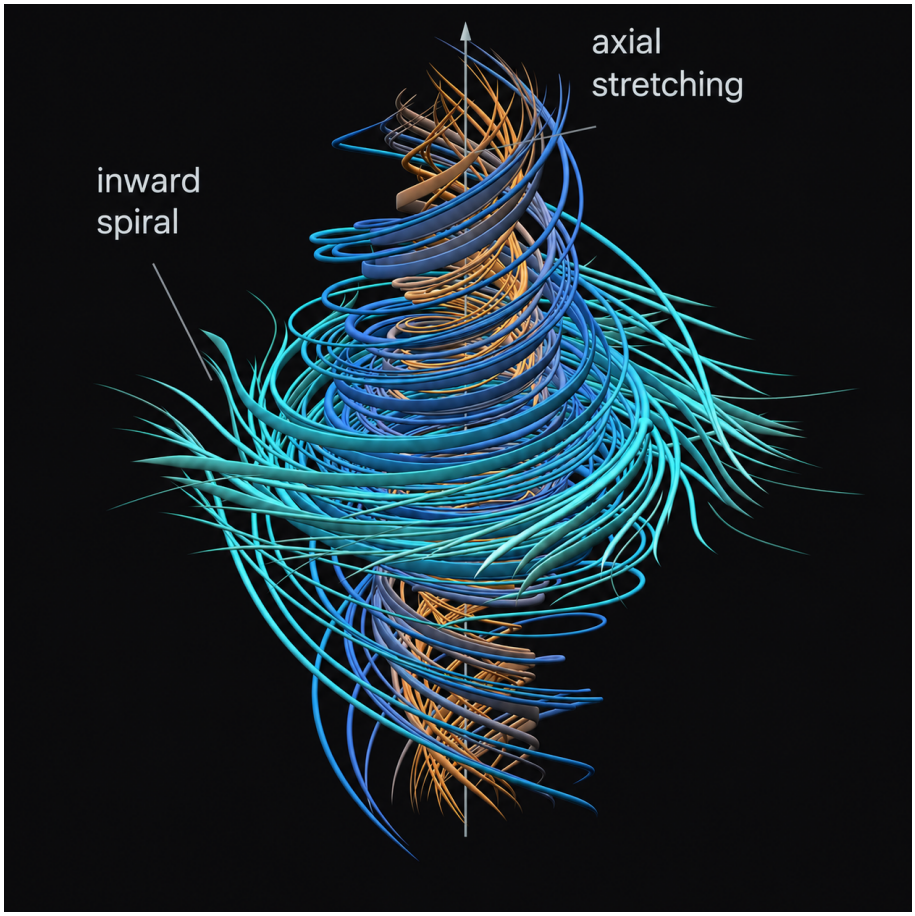

Figure 2: "A snapshot of local incompressible motion. Orange marks faster angular rotation; teal marks slower rotation. Circulating speed also depends on radius. The trajectories show inward spiraling and axial stretching." Image and caption reproduced from OpenAI; caption quoted verbatim. (OpenAI 2026c)

The first term in equation 20 is the rate of change of kinetic energy. The second is viscous dissipation, which takes energy out. The right-hand side is the rate at which the force does work on the fluid.

Since f is smooth and compactly supported, we can use equation 20 to bound the energy, the total work, and the total viscous dissipation up to the blowup time.[10]

The energy bound is on the integral of squared speed, not the maximum speed. The speed can grow without bound in a region that shrinks fast enough. We work out the scaling in section 3.3.2.

3.3.2 The vortex, physically

The motion illustrated in figure 2 is part of a vortex collapsing onto the origin at $t = 1$. Write $\tau = 1 - t$ for the time remaining. In cylindrical coordinates, the fluid spirals inward toward the axis and flows outward along it, upward on one side of a dividing layer near $z = 0$ and downward on the other (OpenAI 2026a).

The spin-up comes from conservation of angular momentum. For a fluid parcel with no torque, angular momentum per unit mass ru_θ is conserved. Decreasing r therefore increases u_θ, like water turning faster as it nears a drain. Viscosity transports angular momentum outward, so the resulting growth depends on the balance between inward transport and viscous loss. Incompressibility requires the incoming fluid to flow out along the axis, allowing the inflow and spin-up to continue.

The core's radial width ℓ_r and axial length ℓ_z both shrink, but at different rates:

$$\ell_r \asymp \tau^{1/2}, \qquad \ell_z \asymp \tau^{1/2-h}, \qquad 0 < h < \tfrac{1}{100}, \tag{21}$$

where $A \asymp B$ means A is bounded above and below by fixed positive multiples of B. Think of a two-sided big-O: the quantities are of the same order, with explicit bounds rather than a loose scaling estimate.

[10] Bound the work rate by $|\int_\Omega f \cdot u| \le \|f(t)\|_2 \|u(t)\|_2$. A Grönwall estimate bounds $E(t)$ uniformly on $[0, 1)$, which also bounds the integrated work. Integrating equation 20 then gives finite total dissipation $\nu \int_0^1 \|\nabla u(t)\|_2^2 \, dt$.

From equation 21, $\ell_r/\ell_z \asymp \tau^h \to 0$, and the core becomes a thinner column as it shrinks. Characteristic speeds scale as $|u_\theta|, |u_z| \asymp \tau^{-1/2-h}$. These are scales across the core, not pointwise estimates: the azimuthal velocity vanishes on the axis. The core's volume is $\asymp \tau^{3/2-h}$, giving kinetic energy $\asymp \tau^{1/2-3h} \to 0$. I.e. the energy in the core goes to zero even though its speed diverges. This is how equations 17 and 18 can hold at the same time.

We can also look at the Reynolds numbers. $Re = U\ell/\nu$ compares the viscous diffusion time ℓ^2/ν with the transport time ℓ/U,

$$\begin{aligned} \mathrm{Re}_\theta &:= \frac{|u_\theta|\ell_r}{\nu} \asymp \tau^{-h} \to \infty, \\ \mathrm{Re}_r &:= \frac{|u_r|\ell_r}{\nu} = O(1). \end{aligned} \tag{22}$$

The azimuthal Reynolds number diverges, so the fluid makes increasingly many turns within a radial diffusion time. The radial Reynolds number stays $O(1)$, so radial transport and diffusion remain comparable. Notice that the radial scale $\tau^{1/2}$ is the diffusive length scale itself.

These power laws are collected in figure 3. Every exponent in it is quoted from equation 21 and equation 22; nothing is simulated or fitted. Two things are worth noticing. The radial width, the axial length, and the core's kinetic energy all fall at essentially the diffusive rate $\tau^{1/2}$, differing only at order h, while the speed rises. And because $h < 1/100$, the anisotropy accumulates extraordinarily slowly: the column has to collapse through a hundred decades of τ before its aspect ratio moves by one.

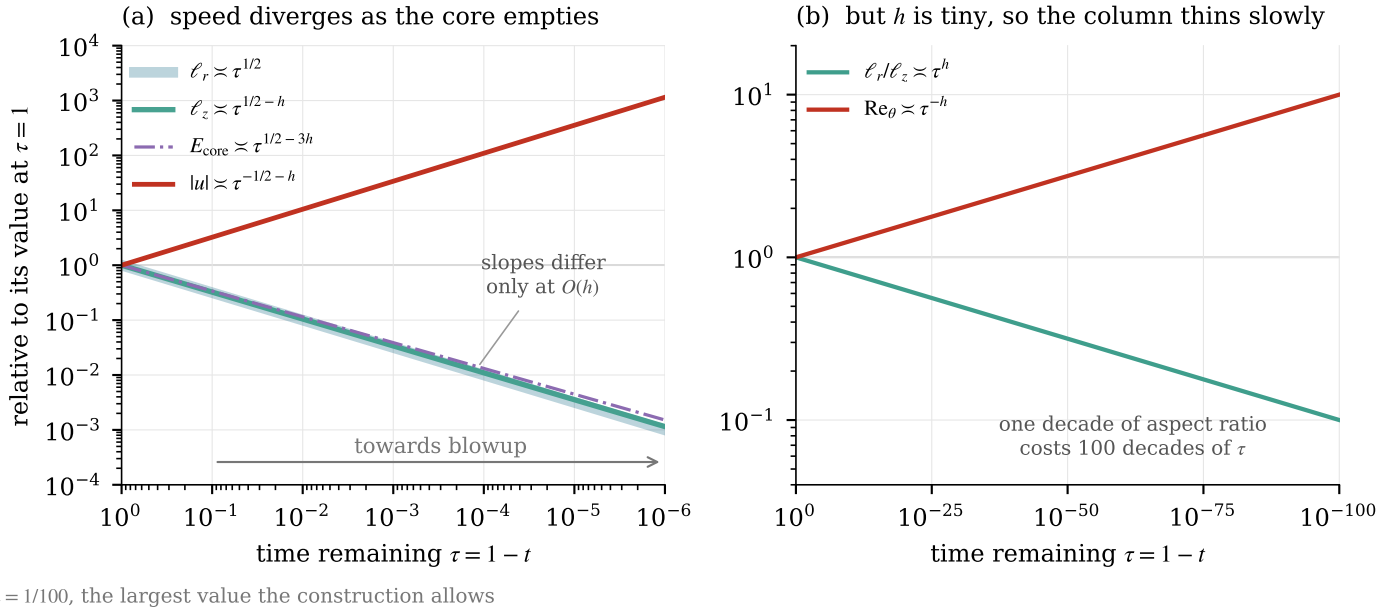

Figure 3: The core's scales as powers of the time remaining, $\tau = 1 - t$, each shown relative to its value at $\tau = 1$. Time runs left to right. Drawn from the exponents in equation 21 and equation 22 with $h = 1/100$, the largest value the construction allows.

3.3.3 Working backwards

The key to these constructions is, naïvely, working backwards: choose a candidate divergence-free u and pressure p, then define the force needed to produce that motion. Define the Navier–Stokes operator

$$\boxed{\mathcal{N}(u,p) := \partial_t u + (u \cdot \nabla)u - \nu\Delta u + \nabla p.} \tag{23}$$

and set $f = \mathcal{N}(u,p)$. The momentum equation in equation 1 then holds by construction.

Here we are constructing the input f for a counterexample; in the usual initial-value problem, f is already given. The hard part: choose u, p so u blows up while the resulting f stays smooth **through** $t = 1$ and obeys the required decay. Rearranging equation 1 alone does not ensure this. Individual terms in the residual can diverge, but their sum and all its derivatives must extend smoothly across the singular time.

Here is a sketch of the iteration.

Let's start with a sort of ansatz in the solution u with some required properties, a "self-similar blowup profile ansatz". Ansatz is German for a guessed form of the solution. Here self-similar means the profile keeps its shape when

lengths and velocities are rescaled using ℓ_r, ℓ_z from equation 21 and the speed scale $\tau^{-1/2-h}$. This describes the leading profile; the corrected solution will not be exactly self-similar. Substitute this ansatz into the operator in equation 23:

$$\boxed{R_0 := \mathcal{N}(u^{(0)}, p^{(0)}).} \tag{24}$$

We join the inner core to a smooth exterior where speed decreases with radius. This leaves a nonzero residual in the annulus between them, which becomes unbounded as $t \uparrow 1$.

We could set $f = R_0$ in equation 24 if we didn't need the force to be nice. The trick is to keep the smooth part as a contribution to f, and then come up with a clever way to pass the bad part into the velocity. I.e. we are **repairing the residual without destroying the blow-up**.

Now the corrections have to keep satisfying the constraints, so it is not completely arbitrary to shove the bad part into them: each one must be divergence-free, and the pressure gets adjusted alongside. So we construct,

$$\boxed{u = u^{(0)} + w_1 + w_2 + w_3 + \cdots} \tag{25}$$

After the first correction,

$$u^{(1)} = u^{(0)} + w_1,$$

define the new residual

$$\boxed{R_1 := \mathcal{N}(u^{(1)}, p^{(1)}).} \tag{26}$$

Choose w_1 so that its nonlinear momentum flux cancels the dangerous part of R_0.[11]

How? That's another clever trick. For a divergence-free w, the nonlinear term is a divergence of a flux,

$$\begin{aligned} (w \cdot \nabla) w &= \nabla \cdot (w \otimes w), \\ (w \otimes w)_{ij} &= w_i w_j, \end{aligned} \tag{27}$$

where $w \otimes w$ is the momentum flux per unit density. Let $\langle \cdot \rangle$ denote an average over the rapid oscillations. Even if $\langle w \rangle = 0$, we can still have $\langle w \otimes w \rangle \neq 0$, because the products of components need not average to zero. So we can choose the oscillations to produce a mean stress that cancels the bad part of the residual:

$$\nabla \cdot \langle w_1 \otimes w_1 \rangle \approx -\text{bad part of } R_0. \tag{28}$$

It doesn't cancel everything exactly. Adding w_1 introduces interactions with the background, as well as its own time derivative and viscous terms. R_1 contains the smooth part we keep for f and these new errors. We need the new errors in equation 26 to be better controlled near $t = 1$.

Then repeat.

Thus the cartoon is

$$\boxed{u^{(0)} \to R_0 \to w_1 \to R_1 \to w_2 \to \cdots} \tag{29}$$

where

$$\begin{aligned} u^{(n)} &= u^{(0)} + \sum_{j=1}^{n} w_j, \\ R_n &= \mathcal{N}(u^{(n)}, p^{(n)}). \end{aligned} \tag{30}$$

The corrections in equation 30 occur on increasingly fine scales.

[11] "Dangerous" here means: fails to extend smoothly through $t = 1$—including its derivatives. A residual can be perfectly bounded and still be dangerous in this sense.

Viscosity acts like a low-pass filter, so how do corrections at increasingly high frequencies survive? The background shear amplifies the seeded pulses before viscosity damps them. We need both enough growth and the required momentum transport, and making these work together is a large part of the proof (OpenAI 2026a).

3.3.4 Taking the limit

$$\boxed{f := \lim_{n\to\infty} R_n = \mathcal{N}(u,p).} \tag{31}$$

Therefore the limit in equation 31 satisfies equation 1 exactly.

To take this limit we need convergence strong enough to differentiate the sum and evaluate the nonlinear term. Before $t = 1$, the velocities, pressures, and the derivatives being used must converge sufficiently well on compact sets. The residual and all its derivatives must also extend smoothly across $t = 1$. Compact support is a separate requirement; smooth convergence alone does not guarantee it.

The trick is that the iteration has been designed so that

$$\boxed{f \in C_c^\infty} \tag{32}$$

is perfectly smooth (and compactly supported), while the resulting velocity nevertheless develops the desired finite-time singularity.[12]

3.3.5 For every positive viscosity

The result holds for every positive viscosity (OpenAI 2026a). One might expect sufficiently large ν to smooth out the flow and prevent blowup. But Navier–Stokes has a scaling symmetry: starting with a solution at $\nu = 1$, define

$$\begin{aligned} u_\nu(x,t) &= \sqrt{\nu}\, u(x/\sqrt{\nu}, t), \\ p_\nu(x,t) &= \nu\, p(x/\sqrt{\nu}, t), \\ f_\nu(x,t) &= \sqrt{\nu} f(x/\sqrt{\nu}, t). \end{aligned} \tag{33}$$

Substituting equation 33 into equation 1, every term picks up the same factor $\sqrt{\nu}$, and the velocity stays divergence-free. So this gives a solution at viscosity ν. Time is unchanged, including the blowup time.

For this construction we can therefore take $\nu = 1$ without loss of generality.[13] Changing ν changes the length scale, and we rescale the velocity and force along with it. The radial Reynolds number stays $O(1)$, as in equation 22.

3.4 Observations

Is C/D just a matter of pushing hard enough? The force is smooth and does a finite amount of work, as we saw in section 3.3.1. The singularity comes from concentrating the motion into a shrinking region.

The iteration in equation 29 requires a delicate balance of cancellations. Would a small perturbation of u_0 or f preserve the singularity?[14] Anandkumar's group is aiming for a stable singularity, though its rigorous stability argument is still unfinished (Anandkumar 2026).

So smoothness and rapid decay of the data are not, once forcing is allowed, sufficient for global regularity. Are there natural assumptions one could write down that would be? Some cases are already known: A/B hold in two dimensions, and in three dimensions with a smallness condition on u_0. For general data, smooth solutions exist at least for a short interval $[0, T)$. Leray's weak solutions exist globally, so losing smoothness does not mean losing every notion of solution (Fefferman 2000, 2–3). Could we find a natural condition between small data and unrestricted data?

How about the missing corner? Would the A/B form of the Navier–Stokes Millennium Prize Problem be true, in which case we cannot provide a construction leading to a blowup, because it cannot happen? OpenAI reports that after the Euler result, it prompted agents with that resolution to work on Navier–Stokes, later combining useful intermediate pieces from different groups (OpenAI 2026c). The absence of an A/B counterexample does not establish anything by itself. But perhaps we can try the other direction: if this iterative procedure cannot work without forcing, could the reason suggest a way to prove global regularity?

[12] How does the blowup in equation 18 rule out a global smooth solution with the same u_0, f? By uniqueness before the blowup. Any such solution must agree with the constructed one on $[0, 1)$, so it has the same unbounded speeds and cannot be smooth through $t = 1$. Hence we get the nonexistence statement in C/D.

[13] This spatial rescaling applies directly on $\mathbb{R}^3$. On the unit torus an arbitrary dilation would change the period. First shrink the compactly supported construction to fit inside a cell, then repeat it periodically (OpenAI 2026a, Corollary 10.6).

[14] We perturb the inputs because fixing both gives a unique solution before blowup. The construction alone does not tell us whether the singularity is stable.

How about discretization, such as solving the equations on a finite mesh using the finite element method (FEM)? A mesh or a finite set of modes cannot resolve indefinitely shrinking spatial scales. Would this construction still leave something recognizable at finite resolution? Would different discretizations give different results?

In a fixed finite-dimensional space all norms are equivalent. So if a scheme preserves an energy bound, it cannot have unbounded speed at that resolution.[15] The constants relating the norms can worsen as we refine the mesh, allowing the maximum speed to grow with resolution. This is why the choice of discretization matters.

The PINN approach in table 2 does not evade this issue by simulating all the way to blowup. It searches numerically for a profile in rescaled coordinates, where the shrinking shape is intended to stay fixed. The further task is to certify a nearby continuum solution and its stability. A finite numerical representation is therefore a starting point for that argument, not itself a simulation of an infinite peak (Anandkumar 2026).

This leads back to physics: what does the construction tell us about a real fluid? We assumed a continuum, incompressibility, and Newtonian viscous stress to derive the equations. Which assumptions would stop being valid as the vortex collapses, and what should replace them?

Perhaps we can speculate using a Wilsonian view. At different scales we might have different effective field theories, each valid down to some shortest length, a UV cutoff. This construction has no such cutoff: the spatial scales become arbitrarily small and the speeds arbitrarily large. A real fluid would encounter molecular structure before $\tau \to 0$, and diverging speeds would invalidate incompressibility.

The "new" physics here could be effects omitted from the model, even if they are already known elsewhere. Is there anything deeper? Could the vortex be interpreted as a topological defect, a quasiparticle, or a critical excitation?[16] Perhaps we can start by asking which neglected effect becomes relevant first, and whether the mechanism survives when we include it.

Could OpenAI's agents help answer these questions? They worked on C/D with a checkable target. Without one, would massive parallelism and long runs still produce useful work? Or would we get many plausible answers and no way to tell which are worth following? We come back to this in section 4.3.3.

4 AI

4.1 Lean and formal verification

Two of the three results here were checked by a machine before human review. This seems to have received less attention than the results themselves, but I think it deserves some explanation (Cook 2026a, 2026b).

Lean is a compiler with a type system that lets us express theorems as types and proofs as programs. Compiling a proof checks that the theorem follows from the axioms. A small kernel checks each inference. Tactics, automation, and libraries produce proof terms that the kernel then checks. To someone familiar with programming, this gives a way to check mathematical reasoning using a type checker.

The history of computer-assisted proofs helps explain why mathematicians want this. Hartnett's book on Lean starts with Hales' proof of the Kepler conjecture (Hartnett 2026). Hales announced it in 1998, then waited years for a dozen referees. According to Hartnett, much of that time was spent waiting for someone to start: working through another person's computation was not what they wanted to spend their careers on. The proof was finally published in 2005, with the referees saying they believed it was correct but could not check it fully. Hales started Flyspeck to formally verify it; the project finished in 2014.[17] Hartnett then follows the development of Lean and `mathlib`, from Leonardo de Moura's work at Microsoft Research to the mathematicians who built on it.

Buckmaster describes his own Euler write-up as something that "can only be described as AI slop." Yet the proof is Lean-verified (Buckmaster 2026b). This gives us an example of a correct proof that is still difficult to read or learn from. Previously, an unreadable proof was also difficult to verify, since verification depended on someone reading it. Formal verification lets us check correctness separately from how well the proof is written.

We still need human review. The kernel checks that a proof follows from its hypotheses, but it cannot tell us whether we wrote down the statement we intended. We also need to check which assumptions the proof depends on. There are two parts here:

1. Does the Lean theorem express Clay C/D? For example, does `divergence` mean classical divergence, and does the convection term mean $(u \cdot \nabla)u$? Does smoothness on $\mathbb{R}^3 \times [0, \infty)$ include the one-sided derivative at $t = 0$?

[15] Finite-dimensionality alone does not prevent blowup: $\dot{y} = y^2$ with $y(0) > 0$ gives $y(t) = y_0/(1 - y_0 t)$, singular at $t = 1/y_0$. The energy bound is essential. Truncating to finitely many modes, obtaining energy bounds, and passing to the limit is also how Leray–Hopf weak solutions are constructed.

[16] These interpretations would need more structure. For Euler, Kelvin's theorem conserves circulation along the flow, but that does not imply quantization. Viscosity removes even that conservation law for Navier–Stokes.

[17] Flyspeck used HOL Light and Isabelle. Lean began separately in 2013 (Hales et al. 2017; de Moura et al. 2015).

Do the force-decay conditions agree with Clay's multi-index conditions? A mistake here could give us a verified proof of a different theorem.

2. What does the proof assume? Check the full dependency closure for `sorry`, custom `axiom` declarations, `native_decide`, or other ways to bypass the intended kernel check. Running `#print axioms` on the final theorem should report at most the three standard axioms, `propext`, `Classical.choice`, and `Quot.sound`. The dependency checks can be automated.

One subtlety is that Lean's functions are total, even for operations that we usually treat as partial. The integral of a non-integrable function is assigned 0, and a derivative is assigned a default value where the function is not differentiable. So a bound like $\int |v|^2 < E$ can hold even when the function's energy should be infinite. To programmers, think of a function silently returning a default value. We need to require integrability alongside the bound; we cannot infer it from the bound alone. This is an example of something a human reviewer has to check in the definitions.

For a nonexistence statement like C/D, the direction of the comparison also matters. If Lean's definition allows fewer global solutions than Clay's, proving nonexistence in Lean would not rule out all the solutions Clay permits. We need to check that every object allowed by Clay is included in the formal definition, as well as checking the constructed example.

This reduces the amount of work a reviewer needs to do. We can compare the definitions with Clay's statement and run the two checks above, without manually reproducing 165 pages of estimates. Perhaps this is a day or two of expert attention instead of a year of refereeing. This is the kind of difficulty illustrated by Hales' proof that formal verification can help with.

OpenAI reports that the Navier–Stokes construction took about 88 hours, followed by another 17 hours for Lean formalization and verification using GPT-6 Astra. That describes a separate formalization stage; it does not tell us every tool used during the preceding search (OpenAI 2026c). Producing the informal proof and translating it faithfully into a formal one are two capabilities. The translation itself takes much of the effort when humans formalize mathematics.

This also depends on `mathlib`. A decade of work on the library makes it possible to reuse mathematical definitions and results when checking a new proof. Without that infrastructure, we would have a 165-page PDF that is difficult to read, from a company with an incentive to promote its result. The formalization community has provided a way for others to check the claim, and I think that contribution deserves more credit.

4.2 Ethics

4.2.1 Did OpenAI take their work?

Probably not. I don't think the evidence supports the stronger claims being made in the coverage.

Buckmaster says: "I have not seen OpenAI's proof. I do not know what their model did, or how. I do not know whether our data was used. I am not accusing anyone of anything." (Buckmaster 2026b) He is raising a concern about the sequence of events. OpenAI states that its researchers and agents did not see the Alpöge–Buckmaster work "through any means" before it became public, and that no specific user data was accessed (OpenAI 2026c).

The original unanswered question was whether their work had been used in training. Buckmaster asked whether the model had access to their Codex sessions, which contained drafts throughout the project. He was told the model does not look up user data, then asked specifically about training and did not get an answer at the time (Buckmaster 2026b). OpenAI initially said it could not rule out that de-identified data from their usage had helped improve its models, although it considered this unlikely (OpenAI 2026c).

On 10 September, OpenAI revised its blog post to say that, following an investigation, Buckmaster's Codex prompts from the preceding two months could not have influenced the system, including through training. It says the internal model was developed through reinforcement learning on top of a previously pretrained model (OpenAI 2026c).

That supersedes the earlier caveat. I think it supports my reading that the original wording was cautious while they checked, rather than an admission of taking someone's work. Checking access logs and tracing possible influence through training are different questions. The later statement answers the second as well, although it is still OpenAI's account of its investigation, not an independently published audit.

Willison pointed out that "used to improve our models" was vague: it could cover several processes, not just training on conversations (Willison 2026). That was a fair question about the initial statement. It should now be read alongside the September 10 update. I still think some of the coverage, such as Hart (2026), was too sensational, but the earlier uncertainty should not be presented as if OpenAI has left it unanswered.

The different approaches also make independent work plausible. In the same week we have OpenAI's multiscale construction with discrete jumps in scale, Anandkumar's group's self-similar ansatz found by neural network and

certified numerically, and Alpöge and Buckmaster's extension of the Córdoba–Martínez-Zoroa iterative forcing program to smooth forcing (Anandkumar 2026). The first two concern unforced Euler and the third forced Euler. The forced and unforced problems require different constructions, even where they share public prior ideas. This makes the suggestion that OpenAI simply took their proof less convincing, though different results alone cannot establish the provenance of training data.

The rumour could explain the timing. OpenAI says it launched the search on 1 September after hearing that two Millennium Prize problems had been resolved (OpenAI 2026c). Knowing that someone is close tells you where to spend compute, even without knowing how their solution works. The forced approach was already public in Córdoba and Martínez-Zoroa's work; Fefferman calls them the heroes of the story (Kakaes 2026).

The rumour was wrong, too. It named Anthropic, but the work was a personal collaboration between Buckmaster and Alpöge, and their result was about forced Euler, not Navier–Stokes.[18] I think this is interesting: even a wrong rumour was enough to get the search started. We come back to this in section 4.3.2.

4.2.2 The pressure to publish

I think there is still an ethical problem here: two groups rushed out work they considered unfinished because of a race they did not choose to enter.

Buckmaster: "Ideally, we would have preferred to spend weeks turning the LLM generated proofs into something readable from the very first page. This level of care is what these problems and the community devoted to these problems deserves." Instead the papers went out as they were, and he apologises for the result in the paper itself (Buckmaster 2026b, 2026a).

Anandkumar's group went public on 7 September, roughly half an hour after circulating the work internally. Tao encouraged immediate release because competing results were appearing, even though the rigorous stability argument was not finished (Anandkumar 2026).

Both groups would have preferred more time. One published an Euler result in a form its author calls slop, and the other released a self-similar profile while certification was still in progress. This affects what the rest of the community gets to read and learn from. OpenAI's announcement schedule put pressure on them whether or not anyone's data was used. A group that can produce results in days can force a much shorter deadline on people who have been working for years.

Priority races have caused problems in mathematics for centuries. AI adds a large difference in how quickly the competing groups can work.

4.2.3 Open problems and credit

Tao worries that "the collection of good, fruitful open problems is now being mined in a non-renewable fashion." (Tao 2026c) The community spends decades developing these problems, while a group with enough compute may solve them very quickly without contributing comparable resources back. Tao is also crowdsourcing a public resource list on AI and mathematics (Tao 2026a). I read this as one attempt to build shared resources alongside the concern about consuming them.

I think this is related to how mathematics assigns credit. The first complete proof receives most of the recognition, while progress toward it receives much less. That makes an open problem something to compete over, and gives little incentive to share unfinished work. The Newton–Leibniz priority dispute is a familiar example from three centuries ago. This "feature" of mathematics survives to the present day.

Think of the drama around the Poincaré conjecture: Perelman's proof, Yau promoting Cao and Zhu's paper as "a complete proof", and the public quarrel that followed (Nasar and Gruber 2006). Perelman turned down both the Fields Medal and the Clay prize. He also thought Hamilton, whose Ricci flow programme he completed, deserved no less credit. We are still fighting over how to credit a proof and the work that made it possible.

The reception of computer-assisted proofs is another example. When Appel and Haken proved the four-colour theorem by computer in 1976, one objection was what would happen if there were a surge of electricity (Barber 2026). Hales' Kepler proof took years to referee and was published with a caveat about verification. The community had difficulty accepting these proofs even though no mathematical flaw had been found.

Perhaps these are related. We tend to reward a complete proof, explained by an identifiable person or group at the end. Incremental progress, formalized results, and the infrastructure supporting them fit less well into that convention. So we undervalue some contributions while fighting over who gets credit for the final result.

[18] On 3 September, Buckmaster mentioned to OpenAI "a rumor … that Anthropic has resolved a major open problem" (Buckmaster 2026b). OpenAI seems to have learned what had actually been proved through the exchanges of 3–6 September, by which time its agents had reached the Navier–Stokes result.

4.2.4 Could open development help?

Some advice now is to keep unpublished mathematics off AI tools and share less of it (Hart 2026). I can understand why an individual would do that. But if everyone responds this way, we get more secrecy and duplicated effort, with the same competition for the first complete proof.

Why not try open development, as in open source software? Work is visible from the first commit, with timestamps and attribution for each contribution. There is a record of what someone did even if another person finishes the project. If mathematical work developed this way, and we gave credit as it progressed, being second would still count for something. Perhaps that would reduce the incentive to rush an unfinished paper out.

Gowers and Tao already tried something along these lines with the Polymath projects. It hasn't become the usual way of doing mathematics, but perhaps the current pressure gives us a reason to try it more widely.

One might say: open development helps whoever can move fastest on public information, which right now is the AI labs. "Develop in the open" might accelerate exactly the extraction Tao is worried about. Worth thinking about whether continuous attribution actually solves that, or just makes the theft legible.

That's why AI labs should be part of the solution. AI agents too often leave out where their ideas came from. Openness gives us a record of contributions; collaborative AI needs to preserve that record and credit the people behind it. Attribution alone does not settle how the rewards should be shared. But if we set up the rewards properly, perhaps people would publish creative ideas they would otherwise keep to themselves. Think of the recognition a young amateur astronomer might get for a discovery. Can you imagine an amateur mathematician contributing an idea to the next Millennium Prize solution, and receiving credit even if an AI finishes the proof?

4.2.5 The danger of "you can buy a proof now"

Using Willison's calculation, OpenAI's output tokens would cost about $15 million at public API prices across all the problems it attempted, including about $6.5 million for Navier–Stokes (Willison 2026; OpenAI 2026c).[19] The Millennium Prize is $1 million, and OpenAI isn't claiming it.

For a frontier AI lab, the proof is worth more than the prize: the publicity and demonstration of capability can justify spending millions. Almost nobody else can afford to do this.

Traditionally, mathematics is one of the few corners of STEM where "everyone is created equal": you don't need big funding to build big experiments to participate. Pen and paper does the job.[20] Not anymore. Whoever has the money now has a large advantage, and that is a concentration of power. Even at an individual level, if I can afford ChatGPT Plus and you can afford Pro, I'm at a disadvantage. Extend that to institutions and countries, and you can see the problem.[21]

What if a mathematical result is valuable enough that you would want to keep it private?

There is a precedent: GCHQ developed public-key cryptography in secret in the 1970s, years before the published discoveries, and kept it classified until 1997 ("James Ellis | GCHQ," n.d.).[22] Disclosing it would have given away the advantage.

Modern cryptography relies on mathematical problems being hard. A practical way to solve them could be worth much more than a Millennium Prize, especially if you kept it secret. P vs NP is on the same list of problems we started with.[23] Could people race to prove theorems and keep them private, much as brokers buy undisclosed security vulnerabilities?

Now this starts to resemble the usual AI race argument: even if the mathematics community doesn't welcome it, labs keep going because someone else might get there first, national security depends on it, blah, blah, blah. I don't agree with that argument, but people at AI labs and in governments may act on it anyway. How do we encourage open development if some results are worth more kept secret?

[19] The estimate uses 300 billion output tokens in total, including 130 billion for Navier–Stokes. Public API prices do not tell us OpenAI's internal inference cost. The estimate also excludes training, staff, and other costs.

[20] Access to libraries, advisors, and time has never been equal. But doing mathematics has required much less capital than building an experiment.

[21] Compute will likely get cheaper, but better-funded groups can still buy more of it. In a subject that rewards being first, that advantage matters.

[22] James Ellis, Clifford Cocks, and Malcolm Williamson developed what they called non-secret encryption between 1970 and 1973. Cocks found essentially RSA and Williamson essentially Diffie–Hellman. Ellis died a month before the public announcement.

[23] RSA relies on the difficulty of factoring, elliptic-curve cryptography on discrete logarithms, and lattice-based cryptography on lattice problems. An efficient constructive proof of P=NP could undermine these assumptions; a proof without a practical algorithm would not have the same immediate effect.

4.3 Future of mathematics

4.3.1 A Deep Blue or AlphaGo moment?

Buckmaster calls this "a Deep Blue-Kasparov moment". He emphasizes how quickly the work can now be done: "the important thing is instead the significance that a mathematician and an LLM model can now do all this work in a month. The significance of this with respect to the way we train students, assign credit, referee, and decide what is worth one human life's attention cannot be understated." (Buckmaster 2026b)

The comparison makes sense as a public demonstration of a capability people did not expect. But chess is a fully specified competitive game, and people continued playing it after Deep Blue. Kasparov also did not have to referee the machine's reasoning. In mathematics someone still has to verify the result, which is why Lean matters in section 4.1.

There is also a difference in what winning means. In chess or Go, a match gives an overall comparison of playing strength. AlphaGo's win was taken as a decisive result across opening theory, endgame, and intuition. Mathematics includes many kinds of work, so I doubt we will have an equally clear moment for the whole field.

Perhaps the better description is a **jagged frontier**. LLM agents have demonstrated superhuman capability in sustaining long calculations, persisting with tedious work, and coordinating thousands of attempts. But these are only some parts of mathematics. They are also the parts where scale and persistence can make a large difference without requiring a new conceptual insight.

4.3.2 Knowing that it can be done

Perhaps knowing that a problem can be solved is already a useful contribution. OpenAI started its search after hearing a rumour, even though the rumour named the wrong group and the wrong problem (section 4.2.1). It was enough to make them spend the compute.

Willison compares this with finding security bugs (Willison 2026). Madhavapeddy opened a public PR fixing a path traversal bug in OCaml's `cohttp`, and saw probes for it within about ten minutes. Given only a rough description of the bug, his own agent produced an exploit in under a minute (Madhavapeddy 2026). Just knowing where to look can be enough.

Could mathematics work the same way? If even a rumour gives others a useful starting point, keeping drafts private may not offer much protection. This is another reason I would prefer the open development discussed in section 4.2.4.

Anthropic described something similar in August. An unreleased model raised the lower bound on the proportion of zeta zeros satisfying the Riemann hypothesis from 41.6% to 67.2%, after 650 failed ideas. The human input was "mostly variants of 'keep going' or 'believe in yourself' " (Anthropic 2026). The person prompting it had no mathematics background.

In both cases, encouragement to try seems to have mattered, without anyone supplying a mathematical idea. How should we credit that?

4.3.3 Scale and parallelism

The results discussed here use explicit constructions of counterexamples. This allows many attempts in parallel, since one successful construction is enough. OpenAI reports roughly 10,000 agents over 88 hours for Navier–Stokes and around 100 agents over about 50 hours for Euler (OpenAI 2026c). A mathematician's time is valuable and limited. We need to invest that time carefully in ideas likely to be fruitful, rather than spend years or decades on one risky calculation. With enough agents, one can afford to pursue more of those possibilities.

Suppose we could coordinate several hundred PhD-level mathematicians to work on the tedious parts of one construction. Also suppose they didn't mind getting little credit if it succeeded, or spending months on a dead end if it failed. Could they match the current generation of AI on this task? I think probably they could. But humans have careers to build and limited time, so organizing this would be difficult. Some of the advantage here comes from the ability to pay for a large amount of coordinated work without those constraints.

Being able to do that on demand is still a major capability. It helps explain why AI can be superhuman at some mathematical tasks while remaining much less capable at others.

The structure of the problem also makes this kind of parallelism possible. Searching for a counterexample allows independent attempts: most can fail, and one success is enough. In Go, decisions within one game depend on one another, so we cannot split winning that game into ten thousand independent tasks in the same way. This difference affects how much extra compute helps. I think coordinating ten thousand agents to produce a useful result is itself a substantial achievement in agentic engineering, with applications beyond mathematics.

A checkable target helps with both parallelism and long runs. We can test different constructions and discard failures. We can also follow one approach for a long time, checking whether each step makes progress. Without that feedback, would more agents just produce more plausible answers, and longer runs go in circles? The modelling questions in section 3.4 seem much harder to check this way.

This shows that massive parallelism worked, but it does not show that it was necessary. Buckmaster and Alpöge obtained a comparable class of result with two people and commercial tools, including Claude and Codex. Anandkumar's group used a small team, neural networks, and interval arithmetic (Buckmaster 2026b; Anandkumar 2026). How much came from scale, how much from model capability, and how much from humans knowing which approach to pursue? We don't yet know, and I wouldn't assume the human contribution was small.

4.3.4 What's still missing

How about the creative insight required for Wiles' proof of Fermat's Last Theorem, connecting modular forms and elliptic curves? Or Scholze's perfectoid spaces, introducing a new object that made previously difficult questions tractable and changed how people understood the field? These reports have not demonstrated that kind of advance.

The constructions do involve creativity. What seems to be missing is the ability to find an abstraction that makes the problem much easier to understand. There is a joke that everything is trivial to a mathematician once they understand it. A good abstraction can have this effect: it explains the theorem and lets us build on it, making many subsequent theorems easier too.

This is also what makes `mathlib` useful. A proof that checks and a definition that simplifies everything built on it require different skills. These systems have demonstrated the former; I haven't seen the latter demonstrated at a comparable level. This is part of why I doubt the AlphaGo comparison: persistence helps finish a calculation, but a useful abstraction changes the work we can do afterwards.

Even if AI never invents something like perfectoid spaces, it could still disrupt how mathematics trains people, assigns credit, and supports careers. It does not need to be better at every part of mathematics to cause those problems.

4.3.5 What should mathematics reward?

Can you become a professor without proving something new? Usually that is what the system rewards. If AI makes new proofs much cheaper to produce, perhaps we need to recognize more of the other work that helps mathematics progress. For example:

- Building libraries like `mathlib`. Choosing definitions at a useful level of abstraction lets other people reuse them. This takes mathematical design, which these systems seem less good at than producing individual proofs. We should reward it as mathematical work, rather than treating it mainly as service.
- Digesting proofs. Someone needs to turn the 165-page argument into something people can learn from: explain the mechanism, find a shorter argument, or work out what else follows. This addresses Tao's concern about getting a result without the understanding that comes from working toward it. It deserves more recognition than a survey-paper citation.
- Re-proving known results. A new proof can improve our understanding even when the theorem itself stays the same.

The four-colour theorem is an example of the last point. Thorup, Thomassen, Kawarabayashi, Mohar, Inoue, and Miyashita posted a new proof in March this year. It is still computer-assisted, but gives an $n \log n$ colouring algorithm instead of n^2 and reveals new structure in planar graphs, with implications for colouring on other surfaces (Barber 2026). The theorem was already known, but we learned something from proving it again. Quanta published that story two days after the Navier–Stokes one; I wonder if that timing was intentional.

4.3.6 What should AI labs do?

The mathematical community needs to reconsider how it assigns credit, and the open development proposed in section 4.2.4 needs AI labs to participate. They also have a responsibility in how they present these results. I would like to see them explain a few things more clearly:

- Why these results matter beyond the prize. Open mathematical problems provide a useful capability benchmark: the answer cannot simply be memorized or guessed, and a formal proof can be checked. Improvements here may also help other tasks. Coordinating ten thousand agents on a problem that can be split into many attempts is another useful result in its own right.
- Which capabilities were demonstrated and which are still missing. The systems are strong at sustained parallel search, but these results have not shown something like inventing a new abstraction. Describing that distinction would make the announcements more credible and help with the reaction to them.

- Who contributed the ideas and infrastructure. The main idea comes from Córdoba and Martínez-Zoroa, as Tao, Buckmaster, and Fefferman have acknowledged. The verification also depends on years of work on `mathlib`. Agents should preserve sources and contributor records as they work, and the announcements should explain which ideas and libraries the result depends on. I think failing to give them enough recognition accounts for much of the dispute.

Both communities could also help fund and recognize the work of understanding these proofs. If AI labs want collaboration with mathematicians, paying someone to read the 165-page proof and explain what we can learn from it would be a useful place to start. Making the Lean proof modular and reusable is another: pay someone to digest it and contribute the reusable parts back to `mathlib`. For now this still needs a human, and the work needs support.

References

Anandkumar, Anima. 2026. "Stable Singularity of the Euler Equations on $\mathbb{R}^3$." *What's New*, September 10. https://terrytao.wordpress.com/2026/09/10/stable-singularity-of-the-euler-equations-on-r3/.

Anthropic. 2026. "Claude's Progress on the Riemann Hypothesis." August 10. https://www.anthropic.com/research/riemann-zeta.

Barber, Gregory. 2026. "The Four-Color Theorem Gets a Rare New Proof." *Quanta Magazine*, September 10. https://www.quantamagazine.org/the-four-color-theorem-gets-a-rare-new-proof-20260910/.

Buckmaster, Tristan. 2026a. "Finite-time blowup with smooth forcing for incompressible porous media, for Boussinesq, and for 3d incompressible Euler." Mastodon post. Mastodon, September 8. https://mastodon.social/@tristanbuckmaster/117233413705701198.

Buckmaster, Tristan. 2026b. "Statement." September 8. https://cims.nyu.edu/~tristanb/statement.pdf.

Clay Mathematics Institute. 2018. *Millennium Prize Description and Rules*. https://www.claymath.org/wp-content/uploads/2022/03/millennium_prize_rules_0.pdf.

Clay Mathematics Institute. n.d. "The Millennium Prize Problems." *Clay Mathematics Institute*. Accessed September 10, 2026. https://www.claymath.org/millennium-problems/.

Cook, John D. 2026a. *Navier-Stokes in the News*. September 8. https://www.johndcook.com/blog/2026/09/08/navier-stokes-in-the-news/.

Cook, John D. 2026b. *The Part of Navier-Stokes No One Is Talking about*. September 9. https://www.johndcook.com/blog/2026/09/09/formal-method-revolution/.

Fefferman, Charles L. 2000. *Existence and Smoothness of the Navier–Stokes Equation*. https://www.claymath.org/wp-content/uploads/2022/06/navierstokes.pdf.

Hales, Thomas, Mark Adams, Gertrud Bauer, et al. 2017. "A Formal Proof of the Kepler Conjecture." *Forum of Mathematics, Pi* 5: e2. https://doi.org/10.1017/fmp.2017.1.

Hart, Robert. 2026. "Mathematicians Want Proof OpenAI Didn't Use Their Work." The Verge, September 10. https://www.theverge.com/ai-artificial-intelligence/993263/where-does-openai-get-mathematics-training-data.

Hartnett, Kevin. 2026. *The Proof in the Code: How a Truth Machine Is Transforming Math and AI*. First. Quanta Books / Farrar, Straus and Giroux.

"James Ellis | GCHQ." n.d. Accessed September 11, 2026. https://www.gchq.gov.uk/person/james-ellis.

Kakaes, Konstantin. 2026. "AI Has Solved One of Math's $1 Million Millennium Prize Problems." *Quanta Magazine*, September 8. https://www.quantamagazine.org/ai-has-solved-one-of-maths-1-million-millennium-prize-problems-20260908/.

Madhavapeddy, Anil. 2026. "Just a Rumour of a Bug Is Enough to Find a Security Exploit These Days." *Front Matter*, August 22. https://doi.org/10.59350/tngsm-6rx23.

Moura, Leonardo de, Soonho Kong, Jeremy Avigad, Floris Van Doorn, and Jakob Von Raumer. 2015. "The Lean

Theorem Prover (System Description)." In *Automated Deduction - CADE-25*, edited by Amy P. Felty and Aart Middeldorp, vol. 9195. Lecture Notes in Computer Science. Springer International Publishing. https://doi.org/10.1007/978-3-319-21401-6_26.

Nasar, Sylvia, and David Gruber. 2006. "Manifold Destiny." Annals of Mathematics. *The New Yorker*, August 21. https://www.newyorker.com/magazine/2006/08/28/manifold-destiny.

OpenAI. 2026a. "Finite Time Blowup for Navier–Stokes." Preprint, September 8. https://cdn.openai.com/pdf/32d9f210-8b73-45e0-91bc-82a30aef8a9a/navier-stokes.pdf.

OpenAI. 2026b. "Finite Time Blowup for the Euler Equation." Preprint, September 8. https://cdn.openai.com/pdf/315b36cd-ec98-4023-8342-93345194ece1/euler.pdf.

OpenAI. 2026c. "On the Navier–Stokes Millennium Prize Problem." OpenAI, September 8. https://openai.com/index/navier-stokes-solution/.

Tao, Terence. 2026a. "Crowdsourcing a List of General Resources on AI and Mathematics." *What's New*, September 10. https://terrytao.wordpress.com/2026/09/10/crowdsourcing-a-list-of-general-resources-on-ai-and-mathematics/.

Tao, Terence. 2026b. "Finite Time Blowup with Smooth Forcing Term for the Incompressible Porous Medium, Boussinesq, and Incompressible Euler Equations." *What's New*, September 8. https://terrytao.wordpress.com/2026/09/07/finite-time-blowup-with-smooth-forcing-term-for-the-incompressible-porous-medium-boussinesq-and-incompressible-euler-equations/.

Tao, Terence. 2026c. "The collection of good, fruitful open problems is now being mined in a non-renewable fashion." Mastodon post. Mastodon, September 8. https://mathstodon.xyz/@tao/117237320796901560.

Willison, Simon. 2026. "On the Navier–Stokes Millennium Prize Problem." Simon Willison's Weblog, September 8. https://simonwillison.net/2026/Sep/8/on-navier-stokes/.