

AI as Computational Mathematician Rather Than Theorem Prover

Claes Johnson

KTH Royal Institute of Technology, Stockholm
claesjohnson@gmail.com

October 7, 2026

Abstract

AI has lately been demonstrated as a theorem prover: it produces analytical arguments and in some cases formalises them in Lean. Its mathematical output includes substantial work in partial differential equations, and almost no conventional numerical PDE — no finite elements, no adaptive computation, no error control, no computational turbulence. This paper asks whether that is the right division of labour.

Numerical PDE already has its computational machinery, and there is no reason for a language model to replace it. The comparative advantage lies one level up: *AI decides what should be computed, and conventional algorithms compute it*. Turbulent flow is the natural test case, because the question that matters there is not whether the detailed solution is unique — it is not — but whether mean-value outputs such as drag and lift are reproducibly computable. That question is open, it is quantitative, and it is more naturally settled by controlled computation than by analytical estimate. We argue that the role available to AI is that of a computational mathematician: designing and running the experiments, interpreting them, and turning observed output uniqueness into quantified and ultimately certified output uniqueness.

1 A gap in the repository

OpenAI has recently demonstrated AI as a mathematical theorem prover, generating analytical arguments and in some cases formalising or verifying them in Lean [1, 2]. The mathematics produced includes hundreds of proof families and substantial work in partial differential equations.

What is conspicuously thin is conventional numerical PDE: finite element and Galerkin methods, time stepping, adaptive computation, a posteriori error control, computational turbulence. The machinery by which partial differential equations are actually solved, and by which essentially every engineering prediction involving them is made, is close to absent.

That could mean the subject is unsuitable for AI. We think it means the opposite, and that the division of labour has been drawn in the wrong place.

2 Numerical PDE already has its machinery

For a system such as the incompressible Navier–Stokes equations, deterministic numerical algorithms already perform the computation efficiently: finite element or Galerkin discretisation,

time stepping, nonlinear iteration, sparse linear algebra, multigrid and preconditioning, adaptive mesh refinement. These are mature, they are fast, and their implementation is automated — the FEniCS project generates compiled code from a variational form, including the adjoint problem required for error control [7].

There is little reason for a language model to replace any of it. Once a numerical problem has been specified, conventional algorithms and high-performance computing should perform the trillions of floating-point operations; a model which attempted to do that arithmetic itself would be slower by many orders of magnitude and less reliable.

The comparative advantage lies above the numerical algorithm:

AI decides what should be computed; conventional algorithms compute it.

The research loop is then

AI hypothesis $\rightarrow$ PDE computation $\rightarrow$ error and stability analysis $\rightarrow$ AI interpretation $\rightarrow$ new computation

in which AI writes and modifies code, chooses numerical experiments, varies parameters, launches ensembles, identifies unexpected behaviour, decides where refinement is required, compares methods, analyses results and formulates the next experiment. It might also discover improved numerical methods — stabilisations, preconditioners, adaptive criteria, reduced models, error estimators. Once discovered and validated, these would become ordinary deterministic algorithms, which is the proper fate of a good method.

3 Turbulence as the test case

Computational turbulence is where this division of labour can be put to work, and the reason is that the interesting question there is not the one analysis has been asking.

Three solution concepts. The history is instructive. A *pointwise* solution of the Euler or Navier–Stokes equations exists and is unstable, so determines nothing, since it does not occur. Leray’s *weak* solution of 1934 [3] exists globally and is not unique, so determines nothing either, since it does not say which. The concept developed in [4] is a weak solution together with *control of the residual*, and what it delivers is different in kind: the flow field is not determined pointwise in space and time, and mean-value outputs are.

Output uniqueness. Suppose U_h and V_h are approximate turbulent solutions computed with different meshes, perturbations or schemes. It is entirely possible that

$$\|U_h - V_h\| \not\rightarrow 0, \quad \text{while for a mean-value output } M, \quad |M(U_h) - M(V_h)| \rightarrow 0.$$

This is *output* or *weak* uniqueness: the detailed turbulent solution need not be uniquely predictable for physically relevant mean quantities to be uniquely predictable. Stabilised Galerkin methods together with primal/dual a posteriori error analysis were developed precisely to investigate it, and the computations indicate stability of mean outputs even where the underlying flow is unstable [4, 5, 6].

That is the phenomenon. Whether it holds in general, and under what conditions, is open.

4 Why an analytical proof need not be the goal

A traditional programme would now seek a theorem establishing this stability. It is worth asking why an analytical proof should be considered superior, because three things argue against it here.

Worst-case constants. Analytical estimates in turbulent flow are worst-case by construction, and the constants they produce grow extravagantly with the Reynolds number. A bound which is true and of size $\exp \exp \exp A$ is not a bound any flow can be said to respect, and no measurement could distinguish it from no bound at all. Computation of the actual dual dynamics may reveal stability constants smaller by many orders of magnitude — and those are the numbers that decide whether an output is determined in practice.

No structure to exploit. Symbolic analysis proceeds by rewriting, and rewriting needs something to grip: a symmetry, a similarity variable, an ansatz. Every construction of a singular solution buys its tractability with one. Turbulence affords none — it is defined by carrying structure at every scale with no organising principle to exploit — and that is not a gap in anyone’s technique but what the word means.

What is actually unresolved. The essential issue is not the absence of a traditional proof. It is *convincing control of the limiting process* $h \rightarrow 0$. That control could itself be computational, and this is the point on which the paper turns.

5 Massive computational experiment

Here AI could do what a human research group cannot, which is to run the investigation at scale.

Thousands of turbulent computations, varying independently the mesh size h , the time step Δt , the viscosity ν , the mesh geometry, the stabilisation, the numerical method, the initial perturbations and the boundary approximation. A systematic search through those computations for observables satisfying

$$M(U_h) \longrightarrow M^*$$

even where the instantaneous solutions differ substantially. Primal and dual problems computed together, so that residuals and a posteriori estimates quantify the uncertainty in each output rather than leaving it to be asserted. Independent methods applied to the same case, to test whether the same output is selected by machinery that shares no code.

This is laborious, it is not deep, and it is exactly the kind of work an AI system is suited to: hypothesis, experiment, interpretation, next experiment, at a rate and breadth no group sustains.

The machinery exists; what is missing is scale. It is worth being concrete about how much of this is already in place, since the proposal would be weak if it required a new method.

The dual computation is implemented. A solver for the *dual linearised problem* runs backward in time on a computed stabilised base flow and returns the stability factor S which controls the output error — with a space–time duality identity, checked under refinement, verifying that the hand-derived dual is the genuine adjoint of the linearised forward operator rather than something resembling it. That is the computation Section 4 appeals to when it says the measured stability constants may be far smaller than the worst-case analytical ones, and it is not a prospect: in one space dimension it has been carried out.

The forward solvers exist likewise, for the configurations at issue: bluff-body flow with slip boundary conditions, compressible and nearly incompressible, in two and three dimensions, running on GPU at 400^3 , together with compressible Euler from the Sod shock tube up to 200^3 .

So neither half of the loop needs inventing. What is missing is the dual problem solved on a *three-dimensional turbulent* base flow rather than a one-dimensional one, and the systematic ensemble — thousands of runs, parameters varied independently, outputs tracked, stability factors recorded. That is a matter of organisation and persistence rather than of method, which is precisely why it is the sort of work to hand to a machine.

The progression available is then from *computational evidence*, to *quantified computational evidence*, and possibly to *certified computational mathematics* — rigorous residual estimates, interval arithmetic, verified numerics, turning parts of the computation into computer-assisted proof. Formal theorem proving need not be the starting point. It may be the destination.

6 Why certification is reachable here

There is a structural reason for optimism about that last step, and it distinguishes the computational route from the symbolic one.

Neither a long computation nor a long proof can be verified step by step by a person: one performs billions of floating-point operations, the other contains more inferences than anyone will read. But they escape that position differently, and only one of them escapes.

A numerical result can be checked *without checking the computation*. Substitute the computed field into the equations and evaluate the residual; weight it by the solution of an adjoint problem and it bounds the error in any output of interest. The decisive feature is that this test is independent of how the field was obtained — it inspects no mesh, no solver, no arithmetic, and would be the same test applied to a candidate that arrived by guess. Nobody therefore needs to verify the individual steps, and nobody does: an error at the three-millionth operation is caught because the residual comes out large. The equation audits the answer, once, wholesale.

A proof admits no such operation. What it produces is a statement rather than an object, and a statement cannot be substituted into anything; its correctness simply *is* the correctness of its steps, which must be checked one at a time. A proof assistant does this mechanically and reliably — and establishes only that the steps follow from the stated definitions, not that the definitions say what was intended, which cannot be mechanised.

So the certification of computational results is not an aspiration borrowed from the symbolic side. It is the native mode of the subject, and the a posteriori error estimate is already a theorem about the exact solution, obtained by computation.

7 Two paradigms

The contrast can be put compactly. The first is currently prominent:

$$\text{AI} \rightarrow \text{analytical proof} \rightarrow \text{formal verification}$$

The second is the one proposed here:

$$\text{AI} \rightarrow \text{numerical PDE} \rightarrow \text{computed solutions} \rightarrow \text{error control} \rightarrow \text{scientific conclusion}$$

In the second, AI functions less as a theorem prover than as a *computational mathematician* or experimental scientist. The numerical solver is the laboratory; AI designs the experiments, controls them, interprets the results, searches for reproducible mathematical structure, and decides what to do next.

Two observations about the comparison, neither of them rhetorical.

The second paradigm already carries its error control; the first does not. A finite element computation returns an output together with a bound on how wrong that output may be, computed from the residual weighted by an adjoint solution. A language model returns an argument with no bound of any kind, and the problem it now faces — an output too long to check and too important to trust blindly — is precisely the one numerical analysis met when machines first outran hand calculation, and answered by computing the error alongside the answer. That answer transfers in principle, and the machinery is numerical in both cases, since a model is itself a numerical object.

And the synthesis is not new. Developing analysis and computation together, rather than treating computation as a service added afterwards, has been pursued as a programme for some twenty-five years under the name *Applied Mathematics: Body and Soul* [8, 9], where the name is the thesis: the body of a mathematical subject is its computational content and the soul its analytical content, and a subject which has lost either is crippled. What is new is a machine capable of operating the loop.

8 The question for turbulence

For turbulent Navier–Stokes flow the scientifically important question may not be whether every detailed weak solution is unique. It is not, and the convex-integration constructions have settled that in the negative. The question is whether physically relevant mean quantities are reproducibly and accurately computable — which is what an aircraft designer asks, what a wind tunnel measures, and what the benchmark workshops test by requiring predictions before the measurements are released [10, 11].

So the challenge for AI becomes:

Can AI turn observed computational output uniqueness into quantitatively controlled — and ultimately certified — computational output uniqueness?

That is a different contribution from producing more analytical proofs, and on the evidence of the subject's history it is the more valuable one. The equations have been available since 1755 and the analytical programme has not delivered a drag coefficient. The computational programme has, and what it lacks is not power but the systematic exploration of its own limiting behaviour — which is exactly the work now within reach.

References

- [1] OpenAI, *Finite time blowup for Navier–Stokes*, preprint, 2026. <https://cdn.openai.com/pdf/32d9f210-8b73-45e0-91bc-82a30aef8a9a/navier-stokes.pdf>
- [2] OpenAI, *Finite time blowup for the Euler equation*, preprint, 8 September 2026, 57 pp. <https://cdn.openai.com/pdf/315b36cd-ec98-4023-8342-93345194ece1/euler.pdf>
- [3] J. Leray, *Sur le mouvement d'un liquide visqueux emplissant l'espace*, Acta Math. **63** (1934) 193–248.
- [4] J. Hoffman and C. Johnson, *Computational Turbulent Incompressible Flow*, Applied Mathematics: Body and Soul, vol. 4, Springer, 2007.
- [5] J. Hoffman, J. Jansson and R. Vilela de Abreu, *Adaptive modeling of turbulent flow with residual based turbulent kinetic energy dissipation*, Comput. Methods Appl. Mech. Engrg. **200** (2011) 2758–2767.
- [6] J. Hoffman, J. Jansson, N. Jansson and R. Vilela de Abreu, *Towards a parameter-free method for high Reynolds number turbulent flow simulation based on adaptive finite element approximation*, Comput. Methods Appl. Mech. Engrg. **288** (2015) 60–74.
- [7] A. Logg, K.-A. Mardal and G. N. Wells (eds.), *Automated Solution of Differential Equations by the Finite Element Method*, Lecture Notes in Computational Science and Engineering 84, Springer, 2012.
- [8] The Body and Soul project, www.bodysoulmath.org.
- [9] K. Eriksson, D. Estep and C. Johnson, *Applied Mathematics: Body and Soul*, vols. 1–3, Springer, 2004.
- [10] J. Jansson, *Adaptive Direct FEM simulation with Unicorn/FEniCS-HPC for CS1*, 5th International Workshop on High Order CFD Methods; Zenodo, 2022. <https://doi.org/10.5281/zenodo.6942599>
- [11] J. Jansson, *Adaptive Euler: validation against NASA High Lift Prediction Workshop 4, and preliminary tail-buffet and transonic-buffet simulations*, presented at IFASD 2026.