AI-Assisted Development of a PDE Solver for CMB Spectral Distortions

Ethan Baker
Physics Department
Boston University
ebaker@bu.edu

Hongwan Liu
Physics Department
Boston University
hongwan@bu.edu

Siddharth Mishra-Sharma
Faculty of Computing & Data Sciences
Physics Department
Boston University
smishras@bu.edu

Abstract

We present SPECTROXIDE, an open-source PDE solver for cosmic microwave background (CMB) spectral distortions ($\sim$14,000 lines of Rust with Python bindings) and, to our knowledge, the first research-grade physics code written primarily by an AI agent with no existing implementation to copy from—developed by human–AI collaboration in $\sim$eight weeks. We document several physics bugs that passed the full test suite but were caught only through human domain expertise, revealing recurring failure modes of AI-generated scientific code: self-calibrated tests, a detailed-balance error, numerical-artifact accumulation, energy mis-routing, and an algebraic error hidden below the signal amplitude. While LLMs dramatically accelerate implementation, domain expertise remains indispensable for correctness today, especially in physics applications. We extensively validate the solver against analytic limits and limited publicly available results, and apply it to derive observational constraints on dark photons as an end-to-end closure test.

1 Introduction

Large language models (LLMs) are increasingly used for code generation, but their application to *scientific computing* remains underexplored. Scientific software has a distinctive property: the output must agree with physical reality (or with known analytic limits), providing a ground truth that most software domains lack. This makes correctness both verifiable and demanding: bugs can be physically plausible yet quantitatively wrong, and an AI can construct convincing justifications for incorrect results.

The CMB is the most precisely measured blackbody in nature [1, 2], and any energy injection in the early universe leaves frequency-dependent imprints (*spectral distortions*) whose shape encodes the injection epoch and mechanism [3–6]. A new generation of experiments (most recently FOSSIL [7]) aims to detect these signals, but computing precise spectral distortion templates requires solving a stiff PDE system where the physical signal is $\sim 10^5$ times smaller than the background. No public code currently solves this equation; the field relies on the private code COSMOTHERM [4] and on precomputed tables derived from it. We chose this problem as a testbed for AI-assisted scientific computing because it is genuinely difficult (the numerics involve near-cancellations between terms that individually dwarf the physical signal, divergent emission rates, and stiff operator coupling), yet the underlying physics is well-understood and scattered across the literature, providing a ground truth to check against.

The solver was developed entirely through a human–AI collaboration using Claude Code powered by Claude Opus 4.6/4.7. The AI handled all implementation, testing, and debugging; the human physicists directed the science, provided validation targets from the literature, and diagnosed subtle bugs. This workflow produced a validated solver in approximately eight weeks of part-time human effort but also revealed several physics bugs, all passing the full test suite and all caught by human

Conference on Physics and AI at Stanford University (PAI 2026)

comparison with published results, that illuminate systematic failure modes of AI-generated scientific code. We offer the resulting workflow and failure-mode analysis as a practical guide for physicists adopting AI tools in their own research.

2 The Solver

SPECTROXIDE solves the photon Boltzmann equation governing how the photon spectrum evolves under three scattering processes [4]:

$$\frac{\partial n}{\partial \tau} = \left.\frac{\partial n}{\partial \tau}\right|_{\text{Compton}} + \left.\frac{\partial n}{\partial \tau}\right|_{\text{DC}} + \left.\frac{\partial n}{\partial \tau}\right|_{\text{BR}} + S(x, z)\,, \tag{1}$$

where n is the photon occupation number, τ is an optical depth coordinate, x is dimensionless frequency, z is cosmological redshift, and S is an external source. Compton scattering redistributes photons in frequency without changing their number, while double Compton (DC) and Bremsstrahlung (BR) create and destroy photons. The distortion's shape records when the energy was injected: very early injections are erased completely (no observable signal); intermediate injections leave a μ-distortion, where scattering reaches a Bose–Einstein spectrum with a nonzero chemical potential but cannot make the extra photons needed to recover a blackbody; and late injections leave a y-distortion, where scattering is too slow even to redistribute the photons in energy [8].

Numerical method. The emission rates diverge as $1/x^3$ at low frequencies, making explicit time-stepping unstable; the Compton term is advanced with Crank–Nicolson and DC/BR with backward Euler schemes, coupled through a Newton iteration [4]. The code is $\sim$14,000 lines of dependency-free Rust with a Python interface. It provides 10+ built-in energy injection scenarios (particle decay, dark matter annihilation, dark photon oscillations, etc.) and accepts user-defined injection histories as Python functions, enabling rapid exploration of new-physics models without modifying the solver. This extensibility is a key advantage of AI-assisted development: adding a new injection scenario becomes a natural-language specification rather than a per-model coding task.

3 AI-Assisted Development

3.1 Development Process

The initial codebase was produced autonomously by the coding agent over four sessions ($\sim$21 hours of active time, spread across two days). After parsing key physics references [4, 8, 9] with parallel agents, the agent produced within one hour an initial commit spanning 15 modules with passing unit tests, covering all major physics processes. Subsequent sessions audited and debugged this code, fixing missing terms and numerical errors.

The project then transitioned to a human-directed loop (Fig. 1): the human specified a physics module and a literature validation target, the agent implemented and tested it, and the human compared the output against published figures and diagnosed discrepancies. Crucially, the human never read the solver code line by line; validation was entirely by comparing output against published results.

Two infrastructure choices proved essential. A *persistent memory file* recorded conventions, known bugs, and decisions across sessions; without it the agent drifted back to old assumptions and undid earlier fixes. *Adversarial review agents*—fresh instances prompted to attack the code—caught real issues the main agent missed, including one genuine physics error the human had overlooked.

3.2 Failure Modes

The most consequential failures were physics bugs that passed the full automated test suite but were caught by human comparison with published results (Table 1). All share a common theme: the code produced output that was *internally consistent* and *superficially plausible*, but *physically wrong*.

3.3 Lessons

These failures motivate concrete practices for AI-assisted scientific computing. The final column of Table 1 lists the diagnostic that catches each individual bug; two of these we adopted as standing

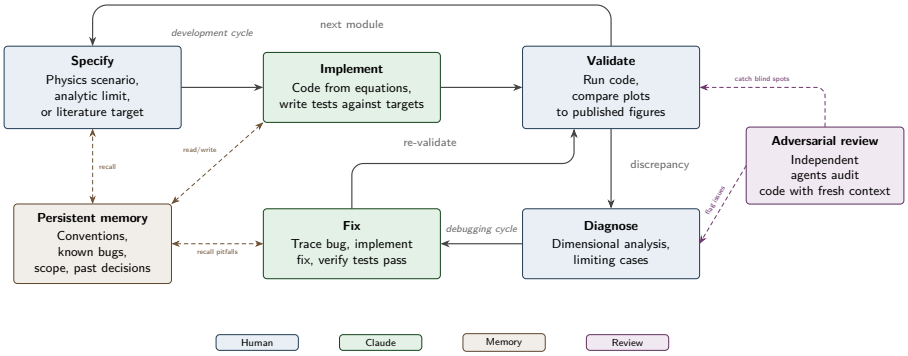

Figure 1: Iterative human–AI development workflow. The outer loop (top) cycles through specifying a physics module, implementing it, and validating against published results. Discrepancies trigger an inner debugging loop (right). Blue: human-driven steps; green: AI-driven steps. A persistent memory file (brown) maintains continuity across sessions. Periodic adversarial review (purple) by independent agents provides fresh-context audits.

Failure mode	Why the full suite passed	What actually catches it
Self-calibrated tests: a spurious factor suppressed Bremsstrahlung by $\sim 10^{11}\times$.	All 375 targets were calibrated to the code's own output, so the suite tested self-consistency, not correctness.	Dimensional analysis of the rate coefficient; test targets derived from analytic or literature values.
Detailed-balance error: The photon absorption rate omitted a physically required factor, giving a factor-of-two error in distortion amplitudes.	No test pinned the absolute amplitude to an independent value.	Compare the μ-to-energy ratio against its analytic value (≈ 1.401).
Artifact accumulation: an ad-hoc energy-leak correction accumulated a spurious signal over hundreds of steps.	Per-step energy error stayed within tolerance; the drift was below the single-step threshold.	A null run: evolve with no injection over the full redshift range and require the distortion to stay zero.
Energy routing: absorbed energy was restored as a uniform temperature shift instead of via electron heating.	Total energy was conserved, so every energy-based test passed; only the spectral *shape* was wrong.	Shape-based tests: decompose Δn into $\mu/y/\Delta T$ and check the shape, not just integrated energy.
Adiabatic cooling: an algebraic slip removed cooling from cosmic expansion.	The $\sim 10^{-9}$ signal was far below the injection amplitudes used in testing.	Reading the paper written by Claude that included the incorrect equation.

Table 1: Physics bugs encountered during SPECTROXIDE's development. All passed the full automated suite and were caught only by comparison with the literature, analytic limits, or human review of agent output.

practices, applied before trusting any numerical output. First, after the self-calibrated-tests bug we required every test target to be derived from an independent source (analytic limit or literature value) *before* implementation rather than read off from code output; we observed noticeably fewer physics bugs slipping through afterward. Second, we required a dimensional check of every rate coefficient before trusting its value—a cheap first-line defense that would have caught the Bremsstrahlung bug immediately, where a spurious factor left a coefficient that must be dimensionless, carrying units of volume. The agent performed these checks reliably when explicitly asked, making dimensional analysis one of the more dependable AI-assisted validation tools. We saw the same failure modes informally with another agent (Codex), where they appeared more frequently than with Claude, though we did not study this systematically.

4 Validation and Application

Validating a PDE solver is hard when none exists to compare against. COSMOTHERM [4], the only other code that solves this problem, is private; only a *table* of its output has been published [8] and

Check	Reference target	Agreement
Early-time injection limit (μ)	analytic, $\mu/(\Delta\rho/\rho) \to 1.401$	few %
Late-time injection limit (y)	analytic, $4y/(\Delta\rho/\rho) \to 1$	sub-%
Energy split into μ vs. y	Ref. [8] fitting formulas	$\lesssim 5\%$
Full spectral shape	CosmoTherm table	sub-%
3 dark-matter injection scenarios	CosmoTherm table	$\lesssim 2\%$
3 stress-test histories	CosmoTherm table	sub-%

Table 2: Validation checks, their reference targets, and the level of agreement. Targets come from analytic theory or CosmoTherm's published output table, available from the CosmoTherm Green's-function distribution [10].

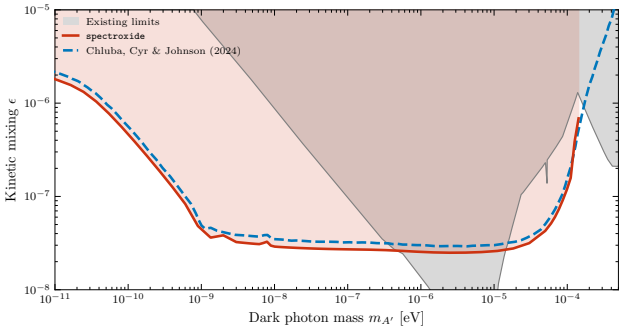

Figure 2: FIRAS 95% CL upper limits on dark photon coupling to photons ϵ as a function of dark photon mass $m_{A'}$. Constraints are derived from PDE solver spectral templates using profile likelihood with floating blackbody temperature. The shaded region is excluded. Our results (red solid) agree with existing constraints (blue dashed; [13]), up to differences in statistical methodology.

distributed publicly [10]. We therefore benchmark against analytic limits and this published table (Table 2); for scenarios it does not cover we use published figures instead. Where we can check, SPECTROXIDE agrees to sub-percent accuracy over the relevant frequencies—rising to a few percent only where the tabulated signal crosses zero—and reproduces all analytic limits.

As an end-to-end closure test, we derive constraints on *dark photons*, a well-motivated Standard Model extension that can convert into ordinary photons [11, 12]. From COBE/FIRAS data [1, 2], we place 95% CL upper limits on the dark photon coupling strength ϵ as a function of mass (Fig. 2), finding $\epsilon \lesssim 3 \times 10^{-8}$, consistent with published results [13, 14] up to differences in statistical methodology (one- vs. two-sided CL). Reproducing an independent published result this way tests the full pipeline rather than yielding a new constraint.

5 Conclusion

SPECTROXIDE demonstrates that an AI agent can produce a validated, research-grade physics solver in a few weeks of part-time effort, with no existing implementation to copy from. The limiting factor was not writing the code but verifying it: every bug in Table 1 passed the full test suite and surfaced only on comparison with analytic limits or published results. They were not crashes or obviously wrong numbers but internally consistent outputs that resembled correct physics, and catching them required a physicist who knew the expected answer to the relevant precision. We expect this to hold: as models become stronger implementers, domain expertise remains essential for correctness, and the dominant cost of AI-assisted scientific computing shifts from implementation to validation. SPECTROXIDE is, to our knowledge, the first open-source PDE solver for this problem; we release the full code, analysis notebooks, and the agent's persistent memory file so the solver and its development process can be reproduced and built on.

References

[1] D. J. Fixsen, E. S. Cheng, J. M. Gales, J. C. Mather, R. A. Shafer, and E. L. Wright. The cosmic microwave background spectrum from the full COBE FIRAS data set. *Astrophys. J.*, 473:576–587, 1996. doi: 10.1086/178173.

[2] D. J. Fixsen. The temperature of the cosmic microwave background. *Astrophys. J.*, 707:916–920, 2009. doi: 10.1088/0004-637X/707/2/916.

[3] R. A. Sunyaev and Ya. B. Zel'dovich. Small-scale fluctuations of relic radiation. *Astrophys. Space Sci.*, 7:3–19, 1970. doi: 10.1007/BF00653471.

[4] J. Chluba and R. A. Sunyaev. The evolution of CMB spectral distortions in the early universe. *Mon. Not. Roy. Astron. Soc.*, 419:1294–1314, 2012. doi: 10.1111/j.1365-2966.2011.19786.x.

[5] J. Chluba. Science with CMB spectral distortions. In *Proceedings of the XLIXth Rencontres de Moriond*, 2014.

[6] J. Chluba et al. New horizons in cosmology with spectral distortions of the cosmic microwave background. *Exp. Astron.*, 51:1515–1554, 2021. doi: 10.1007/s10686-021-09729-5.

[7] Xavier Coulon, Bruno Maffei, and Nabila Aghanim. Towards measurements of CMB spectral distortions. *EPJ Web Conf.*, 293:00012, 2024. doi: 10.1051/epjconf/202429300012.

[8] J. Chluba. Green's function of the cosmological thermalization problem. *Mon. Not. Roy. Astron. Soc.*, 434:352–357, 2013. doi: 10.1093/mnras/stt1025.

[9] J. Chluba, A. Ravenni, and B. Bolliet. Improved calculations of electron–ion bremsstrahlung Gaunt factors for astrophysical applications. *Mon. Not. Roy. Astron. Soc.*, 492:177–194, 2020. doi: 10.1093/mnras/stz3389.

[10] J. Chluba. Cosmological thermalization project. `http://www.chluba.de/CosmoTherm`, 2018. Accessed June 1, 2026.

[11] A. Mirizzi, J. Redondo, and G. Sigl. Microwave background constraints on mixing of photons with hidden photons. *JCAP*, 2009(03):026, 2009. doi: 10.1088/1475-7516/2009/03/026.

[12] Andrea Caputo, Hongwan Liu, Siddharth Mishra-Sharma, and Joshua T. Ruderman. Dark photon oscillations in our inhomogeneous universe. *Phys. Rev. Lett.*, 125:221303, 2020. doi: 10.1103/PhysRevLett.125.221303.

[13] J. Chluba, B. Cyr, and M. C. Johnson. Revisiting dark photon constraints from CMB spectral distortions. *Mon. Not. Roy. Astron. Soc.*, 535:1874–1887, 2024. doi: 10.1093/mnras/stae2464. arXiv:2409.12115.

[14] G. Arsenadze, A. Caputo, X. Gan, H. Liu, and J. T. Ruderman. Shaping dark photon spectral distortions. *JHEP*, 2025:018, 2025. doi: 10.1007/JHEP03(2025)018.