SciCode-Verified: How Benchmark Defects Underestimated the Scientific-Coding Ability of Language Models

Anonymous authors
Paper under double-blind review

Abstract

SciCode is the standard measure of the scientific-coding ability of language models: research-level problems that demand both frontier scientific theory and its implementation as working numerical code. It is a component of the Artificial Analysis Intelligence Index and a standing evaluation in government and national-laboratory suites. Yet its scores have recently plateaued: the strongest 2026 models cluster tightly around 60% subproblem accuracy, and a successor model ties its predecessor. We trace this stagnation to defects in the benchmark itself. A per-problem, domain-expert audit of all 65 test problems uncovers 263 defects; **193** of them, spread across **91%** of the main problems, cause correct, instruction-following solutions to be wrongly rejected—through non-reproducible gold answers, over-tight tolerances, or self-contradictory specifications. Critically, 78% of these score-suppressing defects require specialized physics or mathematics knowledge to detect, not mere clerical proofreading. We corrected every confirmable defect to produce SciCode-Verified. The corrections add only the specifications a well-posed problem requires, repair grading, and tighten the tests that were too lenient; every change is recorded with its justification and independently re-checked by a second domain expert. We re-evaluate twelve frontier model snapshots on the corrected benchmark and find a substantial recovery: subproblem accuracy rises from 45–60% to 84–98%, and main-problem accuracy from 9–26% to 69–92%. State-of-the-art models are far more proficient in scientific coding than SciCode has suggested—the bottleneck was not model capability, but the quality of the evaluation instrument. We release SciCode-Verified with its complete audit trail as the corrected public standard.

1 Introduction

SciCode (Tian et al., 2024) is the standard instrument for measuring the scientific-coding ability of language models. Its 80 research-level problems are drawn from real research across mathematics, physics, chemistry, biology, and materials science, and each one decomposes a research workflow into cumulative subproblems: the model must absorb expert-level scientific background, carry a derivation through, choose an appropriate numerical method, and implement it to research-grade accuracy. As language models are increasingly put to work on frontier scientific research—literature synthesis, derivation, simulation, and experiment design—the need for an instrument of exactly this capability grows with it. SciCode occupies a distinctive niche among widely tracked benchmarks: coding evaluations test software engineering, science evaluations test question answering, and SciCode demands both at once, on genuine research problems whose solutions are working numerical code—and adoption has followed: SciCode is a component of the Artificial Analysis Intelligence Index (Artificial Analysis, 2026a), whose public leaderboard scores 280 model configurations against it (Artificial Analysis, 2026c); government evaluators include it in the UK AI Security Institute's Inspect Evals suite (UK AI Security Institute, 2026); and national laboratories run it as a standing evaluation (Lawrence Berkeley National Laboratory, 2026). Recent model cards and technical reports likewise increasingly evaluate on SciCode alongside established benchmarks (Google DeepMind, 2026; Team et al., 2026; 2025; OpenAI, 2026; ByteDance Seed, 2026; Artificial Analysis, 2026b).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

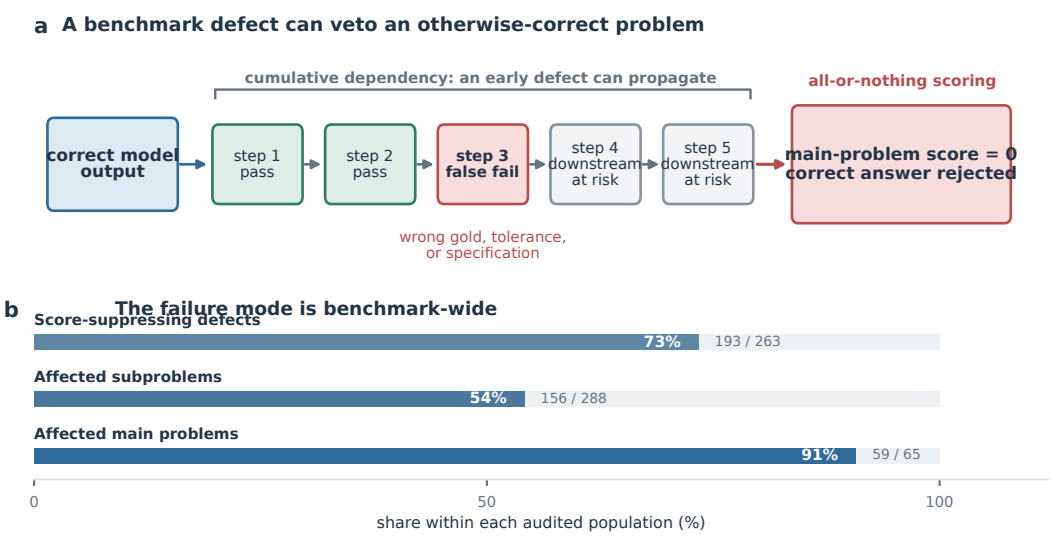


Figure 1: How score-suppressing defects become a benchmark-wide failure mode. **(a)** Because SciCode subproblems are cumulative, an early defect can propagate downstream; because main-problem scoring is all-or-nothing, one false failure can veto an otherwise-correct problem. **(b)** Of the 263 defects identified, 193 (73%) reject correct solutions; these touch 156 of 288 scored subproblems (54%) across 59 of 65 main problems (91%).

For an instrument of this standing, however, its scores behave strangely. Most tellingly, the frontier is numerically indistinguishable: on the public leaderboard, the strongest 2026 models sit within four points of each other—Claude Fable 5, the strongest public model of mid-2026 on the overall Intelligence Index, scores 60%; Gemini 3.1 Pro and Kimi K3 score 59%; GPT-5.6 Sol ties its predecessor GPT-5.5 at 56% (Artificial Analysis, 2026c).[1] An instrument that cannot tell a flagship from its predecessor leaves only two explanations for its low scores: either scientific coding genuinely lies beyond today's models, or the instrument itself is miscalibrated.

We find that much of the gap stems from the benchmark's design and its defects, not from the models. Our domain-expert review of all 65 test problems finds 263 defects. We fix all of them but one[2] to produce SciCode-Verified, the corrected benchmark released with this paper. Two properties of SciCode's design make defects costly here: subproblems form a cumulative chain—step k builds on steps $1{:}k{-}1$—so a defective early step contaminates every step downstream; and scoring is all-or-nothing, so one defective subproblem forfeits the main problem. The confirmed defects are dense and mostly scientific: **193** cause a correct, instruction-following solution to be graded wrong—through non-reproducible or incorrect gold answers, over-tight tolerances, over-specified randomness, or incomplete and self-contradictory specifications—and touch **91%** of the main problems (59 of 65), while **151 of the 193** (78%) are recognizable as defects only with specialist knowledge of the relevant physics or mathematics. Since every model meets the same broken steps, a defect shared by all models becomes a ceiling shared by all models—which is precisely the compression and stagnation observed on the leaderboard. The corrections behind SciCode-Verified stay minimal: each either supplies a constraint that a well-posed problem requires but the prompt left implicit, or repairs grading, and where a test was too lenient we tightened it, so no corrected problem admits a wrong answer (§4). Every change is recorded with its justification and independently re-checked (§A).

Fixing SciCode is hard for a specific reason: most of its defects are scientific rather than clerical—principle-level gaps in the specification as much as numerical errors in gold values and tolerances—so

[1] Artificial Analysis SciCode leaderboard (Artificial Analysis, 2026c), accessed 2026-07-21: Claude Fable 5 60%, Gemini 3.1 Pro 59%, Kimi K3 59%, Muse Spark 1.1 58%; the next nine frontier entries score 53–56%, among them GPT-5.5 and its successor GPT-5.6 Sol, tied at 56%; the 280 listed configurations (multiple settings per model) averaged 42.9% in the 2026-06-02 snapshot.

[2] The one we cannot fix is problem 2 (`Gaussian_Beam_Focus`) itself: its specification fixes no unique answer and so admits no verifiable gold, so we drop the problem rather than ship an arbitrary gold (§2).

both recognizing and correcting them demands domain specialists. A wrong gold value or tolerance need not fail any surface check, and recognizing it—let alone computing the right one—means redoing the physics, problem by problem, across five research fields. Our audit therefore worked at the level of the science rather than the code: an independent reviewer re-derived each problem's targets and stress-tested its specification, and a second, adversarial pass re-verified every flagged defect to eliminate false alarms. Approved changes are recorded with their justification, and each release is mechanically checked against that record (§A). The result, SCICODE-VERIFIED, can now serve SCICODE's original purpose: to judge models' true command of frontier scientific theory and of the code that implements it.

Contributions.

- **The corrected benchmark SCICODE-VERIFIED** (§A): a corrected release of the 64 verifiable test problems, produced so that every correction can itself be audited—each change is recorded with its reason, and the released files are regenerated from that record and mechanically checked against it.
- **A defect taxonomy for SCICODE** (§4): we manually verify 263 defects across 64 of the 65 test problems—each grounded in the audit record and re-checked by an independent adversarial pass—and sort every one along two axes: its *nature* (SCIENTIFIC, needing physics/mathematics judgment to detect, vs SURFACE, a typo, broken cross-reference, or code-interface slip) and its *direction* (rejects a correct solution, hardens a too-lenient test, or neutral cleanup). The resulting cross-tabulation (Table 3) makes the expertise-gated structure of the score suppression legible.
- **A matched before/after re-evaluation of twelve frontier model snapshots** (§3): correction raises subproblem accuracy from the 45–60% they score on the original benchmark to 84–98% and main-problem accuracy to 69–92% (a 29–41 point subproblem gain per model), and can change their relative ranking (single pass@1 runs)—isolating the benchmark's own contribution to the measured score.

2 BACKGROUND: SCICODE AND ITS EVALUATION PROTOCOL

SCICODE (Tian et al., 2024) contains 80 main problems, split into 15 development and 65 test main problems. The 15 development problems ship public ground-truth code for development and few-shot use, so measurement—ours, as in all published results—uses the 65-problem test split. Each subproblem ships a Python function header, a docstring specifying inputs/outputs, optional expert-written background, and domain test cases the candidate solution must pass. Subproblems are solved *cumulatively*: step k may call the gold (reference) functions of steps $1{:}k{-}1$. A main problem counts as solved only if *all* its subproblems pass—an all-or-nothing rule under which a single bad gold value or over-tight tolerance fails the entire problem. We report this as *main-problem* accuracy, and use *step* and *subproblem* interchangeably.

A domain test case can wrongly fail a correct solution in two ways. First, the precomputed gold answer it compares against may itself be wrong, or reproducible by no method—so any correct output disagrees with it. Second, the comparison (a numerical `allclose`) may be too strict: a tolerance tighter than the method warrants, or one that silently demands a convention the prompt never fixed (a sign, a unit, an index origin, a grid, or a random-number order). Either way, a correct, instruction-following solution is liable to be graded wrong.

We audit all 65 test problems. One of them, problem 2 (`Gaussian_Beam_Focus`), is structurally underdetermined: its specification fixes no unique answer, and its gold is reproducible by no method. Unlike every other defect we encounter, it cannot be corrected into a verifiable problem, so we drop it rather than ship an arbitrary gold.

3 RE-EVALUATION ON SCICODE-VERIFIED

Frontier models on SCICODE-VERIFIED. We evaluate twelve frontier model snapshots, listed in Table 1, on both the original benchmark and SCICODE-VERIFIED under a matched with-background harness; only the benchmark data differ. On the original benchmark, they score **45–60%** on subproblems and **9–26%** on main problems. On SCICODE-VERIFIED, the ranges rise to **84–98%** and

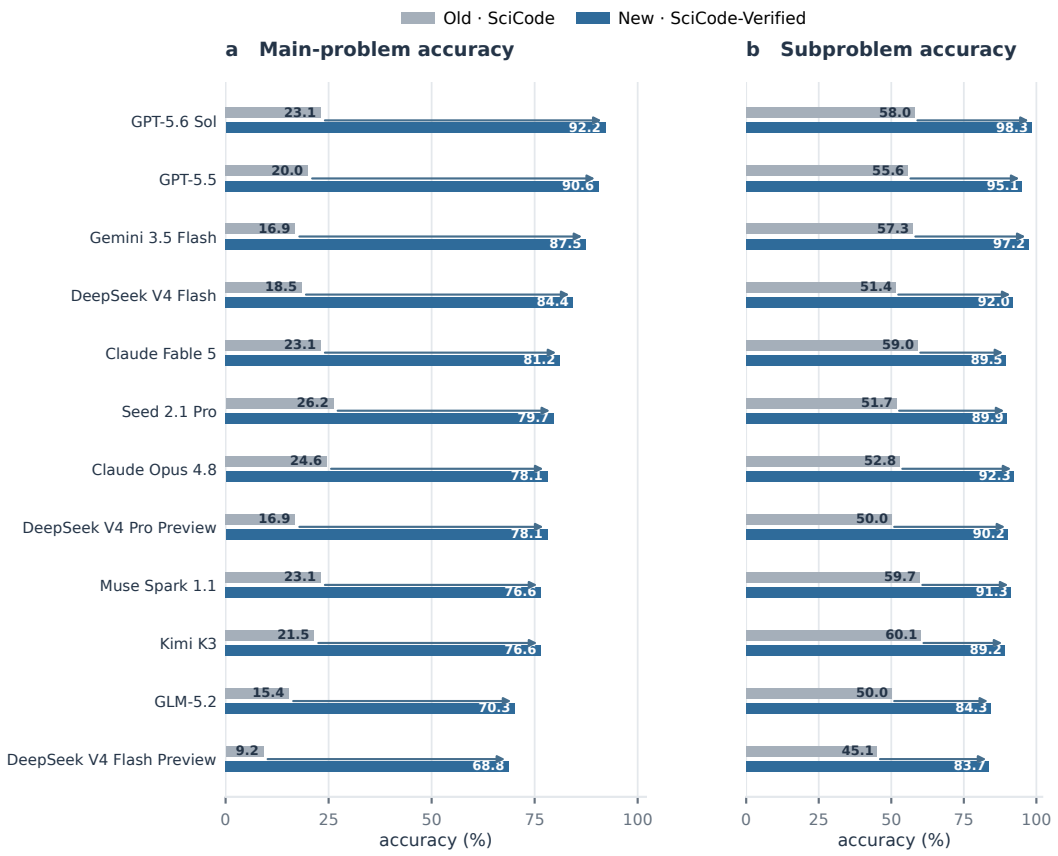


Figure 2: Matched before/after re-evaluation of twelve frontier model snapshots (with background, pass@1, same harness throughout—only the benchmark data differs; Table 1). Within each model, grey and blue bars show accuracy on original SciCode and SciCode-Verified, respectively; the arrow marks the matched old→new change. Correction lifts every model by 53–71 main-problem points **(a)** and 29–41 subproblem points **(b)**. It also restores discrimination: GPT-5.5 and DeepSeek V4 Pro Preview move from two main problems apart to eight, while Seed 2.1 Pro leads the original main-problem score yet ranks sixth on SciCode-Verified. The two DeepSeek V4 Preview rows on SciCode-Verified are the mean of three runs; all other points are single pass@1 runs.

69–92%, respectively (Table 1; Figure 2). Every model gains 29–41 subproblem points and 53–71 main-problem points. Correcting subproblem defects distributed across most problems removes false failures that previously caused entire main problems to be scored as incorrect, producing the larger gains in main-problem accuracy. We analyze the prevalence and effects of these defects in §4.

At the upper end, the corrected benchmark approaches saturation: GPT-5.6 Sol reaches 98.3% subproblem and 92.2% main-problem accuracy. Across the full model set, however, correction does not shift all scores uniformly: it both widens previously compressed gaps and changes model rankings. On the original benchmark, GPT-5.5 passes only two more main problems than DeepSeek V4 Pro Preview; on SciCode-Verified, it passes eight more. Rankings also change: Seed 2.1 Pro ranks first by original main-problem accuracy but sixth on SciCode-Verified. The full pattern is shown in Figure 2 and Table 1.

Following the Artificial Analysis leaderboard (Artificial Analysis, 2026c), we report pass@1. Except for the two corrected-benchmark DeepSeek V4 Preview rows, which average three runs, each entry is a single run. Pass@1 scores vary across runs, so small differences between models, particularly in main-problem accuracy, should not be over-interpreted.

Without the background. SciCode's most realistic setting withholds the expert-written background. With it, GPT-5.5, Gemini 3.5 Flash, and Claude Opus 4.8 solve **92–97%** of subproblems

Table 1: Frontier models on the original vs. corrected benchmark under the *same* harness (with background, pass@1, two-environment OR grading; §3.1)—only the benchmark data differs. *Original* columns are our re-run of the original upstream release on the common 64 problems, with Problem 2 counted as incorrect to report the original 65/288 benchmark denominators. SCICODE-VERIFIED DeepSeek V4 Preview rows (*) are the mean of three runs on an earlier harness revision. Every model gains 29–41 subproblem points after correction.

	Original		SCICODE-VERIFIED	
Model	sub. %	main %	sub. %	main %
GPT-5.6 Sol	58.0	23.1	98.3	92.2
GPT-5.5	55.6	20.0	95.1	90.6
Gemini 3.5 Flash	57.3	16.9	97.2	87.5
DeepSeek V4 Flash	51.4	18.5	92.0	84.4
Claude Fable 5	59.0	23.1	89.5	81.2
Seed 2.1 Pro	51.7	26.2	89.9	79.7
Claude Opus 4.8	52.8	24.6	92.3	78.1
DeepSeek V4 Pro Preview	50.0	16.9	90.2*	78.1*
Muse Spark 1.1	59.7	23.1	91.3	76.6
Kimi K3	60.1	21.5	89.2	76.6
GLM-5.2	50.0	15.4	84.3	70.3
DeepSeek V4 Flash Preview	45.1	9.2	83.7*	68.8*

Table 2: With- vs. no-background accuracy on SCICODE-VERIFIED (%, with-background grading as in Table 1; two-environment OR). Withholding the expert-written background—SCICODE's most realistic setting—costs 10–22 subproblem and 20–36 main-problem points, yet GPT-5.5, Gemini 3.5 Flash, and Claude Opus 4.8 still clear 80–83% of subproblems. GLM-5.2 has no no-background run; DeepSeek V4 Preview rows are the mean of three runs.

	Subproblem %		Main-problem %	
Model	bg	no-bg	bg	no-bg
GPT-5.5	95.1	82.6	90.6	65.6
Gemini 3.5 Flash	97.2	80.5	87.5	51.6
Claude Opus 4.8	92.3	81.9	78.1	57.8
DeepSeek V4 Pro Preview	90.2	68.1	78.1	46.4
DeepSeek V4 Flash Preview	83.7	62.8	68.8	40.6

on SCICODE-VERIFIED, approaching saturation on this metric. Without it, their subproblem accuracy falls to **80–83%** and their main-problem accuracy to **52–66%** (Table 2). Thus, while the with-background benchmark leaves little room on subproblem accuracy, the no-background setting retains meaningful headroom. We analyze the fail-without/pass-with behavior of these three models in §3.1.

3.1 WHAT KNOWLEDGE THE BACKGROUND SUPPLIES

To understand what the background contributes, we analyze 31 subproblems that at least two of three models—GPT-5.5, Gemini 3.5 Flash, and Claude Opus 4.8—fail without the background but pass with it. To identify root causes, we compare the paired generations, reproduce the computations, and trace downstream failures to their originating step. Among model-attributable failures, we identify four recurring root causes.

(i) Reassembling a multi-step derivation. In #32 (optically bound nanoparticle arrays), the background walks from the optical-binding force to the phonon Hamiltonian—trap stiffness $k_i = \alpha E_i^2/w^2$, binding-shifted frequencies $\Omega_i = \sqrt{(k_i + \sum_j k_{ij})/m}$, signed hopping $g_{ij} = -k_{ij}/(2m\sqrt{\Omega_i \Omega_j})$. Without it, the three models produce three *mutually inconsistent* Hamiltonians: Claude Opus 4.8 omits the optical-trap stiffness, GPT-5.5 omits the binding-induced frequency shift, and Gemini 3.5 Flash doubles the trap stiffness. Here, the background provides derivational scaffolding that the models do not reliably reconstruct on their own.

Table 3: Defect taxonomy of the SCICODE audit: 263 defects in 64 of 65 test problems, by *nature* $\times$ *direction*. One defect is the invalid, non-reproducible target in Problem 2, which is recorded in the audit but has no verifiable release target. Of the 193 too-strict-or-wrong defects, 151 (78%) are scientific. The only "too-lenient" entries are weak *tests* we *tightened* ($^{\uparrow}$: stricter, not easier). Provenance: identified in expert review **110**, recovered by the original→corrected diff and confirmed by expert review **153**.

	Direction (per defect)			
Defect nature	Strict/wrong	Neutral	Weak-test$^{\uparrow}$	Total
SCIENTIFIC (needs physics/math judgment to detect)	151	21	6	178
SURFACE (corpus / code / text, trivial)	42	41	2	85
Total	**193**	**62**	**8**	**263**

(ii) Textbook priors overriding the stated formulation. In #54 (SUPG finite elements), the problem statement explicitly specifies a non-standard weak form, yet all three models silently replace it with the *textbook* SUPG/Nitsche formulation. Restating the equations in the background at the point of implementation keeps the models on the stated variant, suggesting that a strong textbook prior overrides information already present in the prompt.

(iii) Specialized domain facts. In #69 (Raman response of a layered electron gas), the models get a domain-specific fact wrong: the backscattering photon momentum transfer is $2k$, not k.

(iv) Near-miss implementation errors. In #79 (Nosé–Hoover chains), the models use the correct reversible-integrator factorization but make isolated bookkeeping errors, such as halving a half-step twice or reusing a stale force array. Here, the background acts as an implementation check rather than supplying a missing method.

Consistent with these being knowledge gaps rather than noise, the no-background penalty is larger for the DeepSeek V4 Preview pair (Table 2): they lose 21–22 subproblem points, compared with 10–17 for the three models analyzed in this diagnostic.

Library-version-robust grading. We grade each saved solution in two pinned NumPy/SciPy environments and count it as correct if it passes either. This recovers 7–9 of 287 subproblems per model that otherwise fail only because of library-version incompatibilities; the rescued sets are nearly identical across models (Appendix B). The rule therefore reduces environment-induced false negatives without favoring any model.

4 WHAT WE CORRECTED: A DEFECT TAXONOMY

We examined and cleaned the test problems one at a time, then added a round of independent adversarial review, to produce SCICODE-VERIFIED. We sort each defect into two broad kinds—SCIENTIFIC, whose identification requires relevant physics or mathematics knowledge, and SURFACE, a corpus, code, or text issue—and by *direction*: whether the original was *too strict or wrong* (failing a correct solution), *too lenient* (a weak test we tightened), or *neither* (a neutral cleanup). Of the 263 defects, **178 are SCIENTIFIC** and **85 are SURFACE**. By direction, **193 of 263** defects (**73%**) are too strict or wrong and therefore suppress measured model accuracy; **8** are too lenient and **62** are neutral. Of these 193 score-suppressing defects, **151** (**78%**) are scientific. They span **156 of 288** scored subproblems across **59 of 65** problems (Table 3). Across the four models included in our subproblem-level attribution analysis, all 164 newly passed subproblems trace to a documented correction.

Scientific failure mechanisms. The 178 scientific defects fall into seven subtypes (Table 4). The largest is unspecified conventions (77), where the gold fixes a choice that the prompt leaves free, followed by wrong gold (33) and RNG-dependent grading (22). The seven examples below show how each subtype distorts evaluation.

Table 4: Sub-structure of the two defect natures (263 defects). SCIENTIFIC subtypes require domain judgment to detect; SURFACE subtypes do not. A full per-defect map is in the appendix.

Subtype	Count	Example
SCIENTIFIC *(178) — needs physics/math judgment to detect*		
Unspecified convention	77	#74 Householder sign sector; index base
Wrong gold (method / sign / symmetry)	33	#22 rotation sign; #13 A_z parity
RNG-dependent grading	22	#13/#46/#50 seeded/exact-scheme → statistical
Non-discriminating test (physics)	14	#5.1 palindromic Lanczos input
Over-tight tolerance	13	#30.3 target is a roundoff fingerprint
Spec↔gold contradiction	9	#8.1 strict boundary vs gold's $\geq$
Invalid / non-reproducible target	10	#2 / #28.1 reproducible by no method
SURFACE *(85) — corpus / code / text, trivial*		
Interface / return-contract mismatch	29	#65 `def tensor()` vs variadic doc
Truncation / garble / dropped constant	20	#15 dropped $\hbar$ mantissa
Broken cross-reference	16	dangling "step ___" pointers
Trivially broken test	12	#66.6 misplaced paren, one-sided tol.
Typo / formatting	8	"fcous" → "focus"

- *Unspecified convention, #74.* Householder QR (Householder, 1958) successively applies reflectors to eliminate entries below the diagonal. In the final column, a 1×1 reflector eliminates nothing: applying it only flips the sign of the last column of Q and the last diagonal entry of R, leaving the product QR unchanged. Applying this reflector and stopping before it are therefore both valid conventions. The prompt did not specify which to use, while the gold accepted only the former; we now state that convention explicitly.
- *Wrong gold (sign error), #22.* The rotation-coefficient recurrence (Gumerov & Duraiswami, 2004) shipped a sign error ($-2 \to +2$) that makes the transfer matrix non-unitary. The corrected sign reproduces the unique unitary Wigner coefficient to 10^{-14}.
- *RNG-dependent grading, #68.* The original diffusion Monte-Carlo test (Metropolis et al., 1953; Reynolds et al., 1982) depended on a particular random-number sequence. We instead check the final statistic, requiring the mean energy to be within 0.05 Ha of the helium ground-state energy.
- *Non-discriminating test, #5.1.* The first Lanczos test (Lanczos, 1950) paired a reflection-symmetric tridiagonal matrix with a palindromic starting vector. Because the matrix preserves reflection symmetry, all Krylov vectors remain in the four-dimensional symmetric subspace, although the test requests six Lanczos vectors. The iteration must therefore break down before exercising all requested steps. We replace the starting vector with an asymmetric one; its first six Krylov vectors are linearly independent, so the same test now exercises the full run.
- *Over-tight tolerance, #30/#67.* For the helium Slater–Jastrow wavefunction (Jastrow, 1955) and the layered electron gas (Jain & Allen, 1985), an `allclose` at `rtol` $= 10^{-15}$ (below double precision) or against a rounding fingerprint fails every correct implementation. We loosen minimally to admit the legitimate method/rounding choice while still rejecting wrong answers.
- *Spec↔gold contradiction, #8.* For Fourier-plane spatial filtering (Goodman, 1996), the prompt requires a strict high-pass boundary, but the gold uses an inclusive ($\geq$) boundary. We regenerate the gold to match the prompt.
- *Invalid or non-reproducible target, #28.1.* For Gaussian-beam propagation through a lens system (Kogelnik & Li, 1966), the target is reproducible by no standard method and contradicts a later subproblem. We regenerate it with the prompt-prescribed method.

The corrections preserve difficulty. To address these defects, we added necessary conventions, corrected erroneous gold answers, and adopted more appropriate verification criteria. These changes prevent valid solutions from being misgraded, allowing models to achieve higher scores without reducing the scientific difficulty of the problems.

5 Related work

Benchmark-verification work has addressed several kinds of defects. UTBoost (Yu et al., 2025) strengthens overly permissive SWE-bench (Jimenez et al., 2024) tests against erroneous patches, while SWE-bench Verified (OpenAI, 2024) human-validates a 500-task subset. Other work corrects defects that suppress accuracy: label errors in factual-consistency benchmarks (Nahum et al., 2025), wrong golds in HumanEval (Liu et al., 2023), and defects in data-engineering tasks (Zanoli et al., 2026). MMLU-Redux (Gema et al., 2025) and Platinum Benchmarks (Vendrow et al., 2025) further show that residual label noise matters near saturation. SciCode contains both kinds of problems: some tests are too weak, while incorrect gold values, over-tight tolerances, and unspecified numerical conventions reject valid solutions.

SciCode's issue tracker documents similar grading defects, including over-strict tolerances[3] and gold solutions that fail their own tests.[4]

6 Limitations

The development split releases gold answers and ground-truth code and is therefore not used for model evaluation. Our audit and corrections cover only the 65-problem test split used for evaluation; all 263 identified defects come from this split. Pass@1 scores exhibit run-to-run variation. Because many models are already near saturation on SciCode-Verified, small score differences among them are more sensitive to this variation. Our reported subproblem accuracies on the original benchmark also differ somewhat from those on the Artificial Analysis leaderboard (Artificial Analysis, 2026c). Although the audit underwent multiple rounds of review, all reviews were conducted by the authors and therefore do not constitute external blind validation.

7 Conclusion

SciCode's ability to distinguish frontier models has been limited by defects in the benchmark itself. Our audit of all 65 test problems identifies 263 defects; we correct every repairable defect and exclude one problem that cannot be made verifiable. Re-evaluating twelve frontier model snapshots on SciCode-Verified, we find that subproblem accuracy rises from 45–60% to 84–98% and main-problem accuracy from 9–26% to 69–92%. We release the corrected benchmark together with a complete record of its defects and corrections to support fair and reproducible evaluation of scientific-coding ability.

SciCode-Verified evaluates single-problem scientific coding rather than long-horizon agentic workflows involving tool use, environment interaction, and iterative development. Developing reliable benchmarks for such workflows remains an important direction for future work.

Reproducibility statement

The anonymous supplementary material contains the corrected benchmark, evaluation harness, manifests, and audit trail needed to reproduce the results. The anonymous artifact is available at `https://anonymous.4open.science/r/scicode-verified-iclr2027-DE54/README.md`. The released manifest binds the data by checksum; original-benchmark runs use the same harness with the upstream text and targets. All scores are pass@1 collected in June–July 2026; all runs are single-sample except the SciCode-Verified DeepSeek V4 Preview rows, which average three. API models were queried with reasoning enabled and provider-default sampling. Failed API calls were retried and per-step outputs and grading verdicts were cached.

AI Use Disclosure

We used generative AI tools to assist with literature discovery, drafting and editing prose, and preparing analysis scripts and figures. The authors reviewed and verified all AI-assisted content,

[3] `https://github.com/scicode-bench/SciCode/issues/8`

[4] `https://github.com/scicode-bench/SciCode/issues/43`

including scientific claims, numerical calculations, code, and figures, and take responsibility for the final paper and released artifacts.

REFERENCES

Sadao Adachi. Gaas, alas, and $al_xga_{1-x}as$: Material parameters for use in research and device applications. *Journal of Applied Physics*, 58(3):R1–R29, Aug 1985. ISSN 1089-7550. doi: 10.1063/1.336070. URL http://dx.doi.org/10.1063/1.336070.

D.J. Adams. Grand canonical ensemble monte carlo for a lennard-jones fluid. *Molecular Physics*, 29(1):307–311, January 1975. ISSN 1362-3028. doi: 10.1080/00268977500100221. URL http://dx.doi.org/10.1080/00268977500100221.

Hans C. Andersen. Molecular dynamics simulations at constant pressure and/or temperature. *The Journal of Chemical Physics*, 72(4):2384–2393, February 1980. ISSN 1089-7690. doi: 10.1063/1.439486. URL http://dx.doi.org/10.1063/1.439486.

S. Arimoto. An algorithm for computing the capacity of arbitrary discrete memoryless channels. *IEEE Transactions on Information Theory*, 18(1):14–20, January 1972. ISSN 1557-9654. doi: 10.1109/tit.1972.1054753. URL http://dx.doi.org/10.1109/TIT.1972.1054753.

Artificial Analysis. Artificial analysis intelligence benchmarking methodology. https://artificialanalysis.ai/methodology/intelligence-benchmarking, 2026a. Accessed 2026-07-18. Intelligence Index v4.1: nine evaluations, SciCode weighted 8%.

Artificial Analysis. Muse spark 1.1: Meta gains 8 Intelligence Index points in three months. https://artificialanalysis.ai/articles/muse-spark-1-1-everything-you-need-to-know, 2026b. Launch analysis, 2026-07-10; SciCode 52%→58%. Accessed 2026-07-18.

Artificial Analysis. SciCode benchmark leaderboard. https://artificialanalysis.ai/evaluations/scicode, 2026c. Accessed 2026-07-21 (chart values are rounded integers). Top entries: Fable 5 60%, Gemini 3.1 Pro 59%, Kimi K3 59%, Muse Spark 1.1 58%; next nine frontier entries 53–56%. 280-configuration average 42.9% as of 2026-06-02 — recheck before submission.

Charles H. Bennett, David P. DiVincenzo, John A. Smolin, and William K. Wootters. Mixed-state entanglement and quantum error correction. *Physical Review A*, 54(5):3824–3851, November 1996. ISSN 1094-1622. doi: 10.1103/physreva.54.3824. URL http://dx.doi.org/10.1103/PhysRevA.54.3824.

H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak. Molecular dynamics with coupling to an external bath. *The Journal of Chemical Physics*, 81(8):3684–3690, October 1984. ISSN 1089-7690. doi: 10.1063/1.448118. URL http://dx.doi.org/10.1063/1.448118.

Blox Bloxham, Hyunseok Lee, and Jeff Gore. Biodiversity is enhanced by sequential resource utilization and environmental fluctuations via emergent temporal niches. *PLOS Computational Biology*, 20(5):e1012049, May 2024. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1012049. URL http://dx.doi.org/10.1371/journal.pcbi.1012049.

J. E. Bowers and C. A. Burrus. Ultrawide-band long-wavelength p-i-n photodetectors. *Journal of Lightwave Technology*, 5(10):1339–1350, 1987. doi: 10.1109/JLT.1987.1075419.

William L. Briggs, Van Emden Henson, and Steve F. McCormick. *A Multigrid Tutorial, Second Edition*. Society for Industrial and Applied Mathematics, January 2000. ISBN 9780898719505. doi: 10.1137/1.9780898719505. URL http://dx.doi.org/10.1137/1.9780898719505.

Alexander N. Brooks and Thomas J.R. Hughes. Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32(1-3):199–259, September 1982. ISSN 0045-7825. doi: 10.1016/0045-7825(82)90071-8. URL http://dx.doi.org/10.1016/0045-7825(82)90071-8.

W. R. Busing and H. A. Levy. Angle calculations for 3- and 4-circle x-ray and neutron diffractometers. *Acta Crystallographica*, 22(4):457–464, April 1967. ISSN 0365-110X. doi: 10.1107/s0365110x67000970. URL http://dx.doi.org/10.1107/S0365110X67000970.

ByteDance Seed. Seed2.1 officially released: Advancing AI productivity. https://seed.bytedance.com/en/blog/seed2-1-officially-released-advancing-ai-productivity, 2026. Release announcement, 2026-06-23. Accessed 2026-07-18.

Xinan Chen, Stefano Chessa, Ian George, Felix Leditzky, and Eric Chitambar. Capacities of entanglement distribution from a central source, 2024. URL https://arxiv.org/abs/2411.04977.

S.W. Corzine, R.H. Yan, and L.A. Coldren. A tanh substitution technique for the analysis of abrupt and graded interface multilayer dielectric stacks. *IEEE Journal of Quantum Electronics*, 27(9): 2086–2090, 1991. ISSN 0018-9197. doi: 10.1109/3.135163. URL http://dx.doi.org/10.1109/3.135163.

R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen differenzengleichungen der mathematischen physik. *Mathematische Annalen*, 100(1):32–74, December 1928. ISSN 1432-1807. doi: 10.1007/bf01448839. URL http://dx.doi.org/10.1007/BF01448839.

J. Crank and P. Nicolson. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. *Mathematical Proceedings of the Cambridge Philosophical Society*, 43(1):50–67, January 1947. ISSN 1469-8064. doi: 10.1017/s0305004100023197. URL http://dx.doi.org/10.1017/S0305004100023197.

Ernest R. Davidson. The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices. *Journal of Computational Physics*, 17(1):87–94, January 1975. ISSN 0021-9991. doi: 10.1016/0021-9991(75)90065-0. URL http://dx.doi.org/10.1016/0021-9991(75)90065-0.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile van Krieken, and Pasquale Minervini. Are we done with mmlu?, 2025. URL https://arxiv.org/abs/2406.04127.

Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, December 1977. ISSN 1541-5740. doi: 10.1021/j100540a008. URL http://dx.doi.org/10.1021/j100540a008.

Joseph W. Goodman. Introduction to fourier optics, second edition. *Optical Engineering*, 35(5): 1513, May 1996. ISSN 0091-3286. doi: 10.1117/1.601121. URL http://dx.doi.org/10.1117/1.601121.

Google DeepMind. Gemini 3.1 pro model card. https://deepmind.google/models/model-cards/gemini-3-1-pro/, 2026. Published 2026-02-19, accessed 2026-07. Reports SciCode (scientific research coding) in its main evaluation table.

Nail A. Gumerov and Ramani Duraiswami. Recursions for the computation of multipole translation and rotation coefficients for the 3-d helmholtz equation. *SIAM Journal on Scientific Computing*, 25(4):1344–1381, January 2004. ISSN 1095-7197. doi: 10.1137/s1064827501399705. URL http://dx.doi.org/10.1137/S1064827501399705.

F. D. M. Haldane. Model for a quantum hall effect without landau levels: Condensed-matter realization of the "parity anomaly". *Physical Review Letters*, 61(18):2015–2018, October 1988. ISSN 0031-9007. doi: 10.1103/physrevlett.61.2015. URL http://dx.doi.org/10.1103/PhysRevLett.61.2015.

Paul Harrison and Alex Valavanis. *Quantum Wells, Wires and Dots: Theoretical and Computational Physics of Semiconductor Nanostructures*. Wiley, 4 edition, 2016. ISBN 9781118923368. doi: 10.1002/9781118923337.

D. R. Hartree. The wave mechanics of an atom with a non-coulomb central field. part i. theory and methods. *Mathematical Proceedings of the Cambridge Philosophical Society*, 24(1):89–110, January 1928. ISSN 1469-8064. doi: 10.1017/s0305004100011919. URL http://dx.doi.org/10.1017/S0305004100011919.

Kristjan Haule. Schrödinger equation for hydrogen atom. Lecture notes, Physics 509: Computational Physics, Rutgers University. https://www.physics.rutgers.edu/~haule/509/04_Scipy_Hydrogen_atom.html, n.d. Accessed 2026-07-17.

Alston S. Householder. Unitary triangularization of a nonsymmetric matrix. *Journal of the ACM*, 5(4):339–342, October 1958. ISSN 1557-735X. doi: 10.1145/320941.320947. URL http://dx.doi.org/10.1145/320941.320947.

T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39-41):4135–4195, October 2005. ISSN 0045-7825. doi: 10.1016/j.cma.2004.10.008. URL http://dx.doi.org/10.1016/j.cma.2004.10.008.

A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4-5):411–430, June 2000. ISSN 0893-6080. doi: 10.1016/s0893-6080(00)00026-5. URL http://dx.doi.org/10.1016/S0893-6080(00)00026-5.

Jainendra K. Jain and Philip B. Allen. Dielectric response of a semi-infinite layered electron gas and raman scattering from its bulk and surface plasmons. *Physical Review B*, 32(2):997–1004, July 1985. ISSN 0163-1829. doi: 10.1103/physrevb.32.997. URL http://dx.doi.org/10.1103/PhysRevB.32.997.

Robert Jastrow. Many-body problem with strong forces. *Physical Review*, 98(5):1479–1484, 1955. doi: 10.1103/PhysRev.98.1479.

Francis A. Jenkins and Harvey E. White. *Fundamentals of Optics*. McGraw-Hill, New York, 4 edition, 1976. ISBN 9780070323308.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues?, 2024. URL https://arxiv.org/abs/2310.06770.

I. Kelson and A.A. Hardy. Strongly pumped fiber lasers. *IEEE Journal of Quantum Electronics*, 34 (9):1570–1577, 1998. ISSN 0018-9197. doi: 10.1109/3.709573. URL http://dx.doi.org/10.1109/3.709573.

Sumeet Khatri, Kunal Sharma, and Mark M. Wilde. Information-theoretic aspects of the generalized amplitude-damping channel. *Physical Review A*, 102(1), July 2020. ISSN 2469-9934. doi: 10.1103/physreva.102.012401. URL http://dx.doi.org/10.1103/PhysRevA.102.012401.

A. M. Knapp, E. J. Walker, and T. W. Baumgarte. Illustrating stability properties of numerical relativity in electrodynamics. *Physical Review D*, 65(6), Feb 2002. ISSN 1089-4918. doi: 10.1103/physrevd.65.064031. URL http://dx.doi.org/10.1103/PhysRevD.65.064031.

H. Kogelnik and T. Li. Laser beams and resonators. *Applied Optics*, 5(10):1550–1567, 1966. doi: 10.1364/AO.5.001550.

Aleksey N. Kolmogorov and Vincent H. Crespi. Registry-dependent interlayer potential for graphitic systems. *Physical Review B*, 71(23), June 2005. ISSN 1550-235X. doi: 10.1103/physrevb.71.235415. URL http://dx.doi.org/10.1103/PhysRevB.71.235415.

C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45(4):255, October 1950. ISSN 0091-0635. doi: 10.6028/jres.045.026. URL http://dx.doi.org/10.6028/jres.045.026.

Lawrence Berkeley National Laboratory. Cborg SciCode benchmark. https://cborg.lbl.gov/bench_scicode/, 2026. Accessed 2026-07.

Peter D. Lax. Weak solutions of nonlinear hyperbolic equations and their numerical computation. *Communications on Pure and Applied Mathematics*, 7(1):159–193, February 1954. ISSN 1097-0312. doi: 10.1002/cpa.3160070112. URL http://dx.doi.org/10.1002/cpa.3160070112.

G. Lehmann and M. Taut. On the numerical calculation of the density of states and related properties. *physica status solidi (b)*, 54(2):469–477, December 1972. ISSN 1521-3951. doi: 10.1002/pssb.2220540211. URL http://dx.doi.org/10.1002/pssb.2220540211.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation, 2023. URL https://arxiv.org/abs/2305.01210.

Shengyan Liu, Zhang-qi Yin, and Tongcang Li. Prethermalization and nonreciprocal phonon transport in a levitated optomechanical array. *Advanced Quantum Technologies*, 3(3), January 2020. ISSN 2511-9044. doi: 10.1002/qute.201900099. URL http://dx.doi.org/10.1002/qute.201900099.

Robert MacArthur. Species packing and competitive equilibrium for many species. *Theoretical Population Biology*, 1(1):1–11, May 1970. ISSN 0040-5809. doi: 10.1016/0040-5809(70)90039-0. URL http://dx.doi.org/10.1016/0040-5809(70)90039-0.

R. Mannella. Quasisymplectic integrators for stochastic differential equations. *Physical Review E*, 69 (4), April 2004. ISSN 1550-2376. doi: 10.1103/physreve.69.041107. URL http://dx.doi.org/10.1103/PhysRevE.69.041107.

Glenn J. Martyna, Mark E. Tuckerman, Douglas J. Tobias, and Michael L. Klein. Explicit reversible integrators for extended systems dynamics. *Molecular Physics*, 87(5):1117–1157, April 1996. ISSN 1362-3028. doi: 10.1080/00268979600100761. URL http://dx.doi.org/10.1080/00268979600100761.

P.M. McIlroy, A. Kurobe, and Y. Uematsu. Analysis and application of theoretical gain curves to the design of multi-quantum-well lasers. *IEEE Journal of Quantum Electronics*, 21(12): 1958–1963, December 1985. ISSN 1558-1713. doi: 10.1109/jqe.1985.1072606. URL http://dx.doi.org/10.1109/JQE.1985.1072606.

W. L. McMillan. Ground state of liquid he^4. *Physical Review*, 138(2A):A442–A451, April 1965. ISSN 0031-899X. doi: 10.1103/physrev.138.a442. URL http://dx.doi.org/10.1103/PhysRev.138.A442.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953. doi: 10.1063/1.1699114.

Pilkyung Moon and Mikito Koshino. Energy spectrum and quantum hall effect in twisted bilayer graphene. *Physical Review B*, 85(19), May 2012. ISSN 1550-235X. doi: 10.1103/physrevb.85.195458. URL http://dx.doi.org/10.1103/PhysRevB.85.195458.

Omer Nahum, Nitay Calderon, Orgad Keller, Idan Szpektor, and Roi Reichart. Are llms better than reported? detecting label errors and mitigating their effect on model performance, 2025. URL https://arxiv.org/abs/2410.18889.

B. V. Noumerov. A method of extrapolation of perturbations. *Monthly Notices of the Royal Astronomical Society*, 84(8):592–602, June 1924. ISSN 1365-2966. doi: 10.1093/mnras/84.8.592. URL http://dx.doi.org/10.1093/mnras/84.8.592.

P. J. J. O'Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, E. Jeffrey, E. Lucero, A. Megrant, J. Y. Mutus, M. Neeley, C. Neill, C. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. C. White, P. V. Coveney, P. J. Love, H. Neven, A. Aspuru-Guzik, and J. M. Martinis. Scalable quantum simulation of molecular energies. *Physical Review X*, 6(3), July 2016. ISSN 2160-3308. doi: 10.1103/physrevx.6.031007. URL http://dx.doi.org/10.1103/PhysRevX.6.031007.

OpenAI. Introducing SWE-bench verified. https://openai.com/index/introducing-swe-bench-verified/, 2024. OpenAI blog post.

OpenAI. Introducing GPT-5.5. https://openai.com/index/introducing-gpt-5-5/, 2026. Published 2026-04, accessed 2026-07. Names SciCode among the ten evaluations composing the Artificial Analysis Intelligence Index.

J. R. Oppenheimer and G. M. Volkoff. On massive neutron cores. *Physical Review*, 55(4):374–381, February 1939. ISSN 0031-899X. doi: 10.1103/physrev.55.374. URL http://dx.doi.org/10.1103/PhysRev.55.374.

Peter J. Reynolds, David M. Ceperley, Berni J. Alder, and William A. Lester. Fixed-node quantum monte carlo for moleculesa) b). *The Journal of Chemical Physics*, 77(11):5593–5603, December 1982. ISSN 1089-7690. doi: 10.1063/1.443766. URL http://dx.doi.org/10.1063/1.443766.

Thomas D. Schneider, Gary D. Stormo, Larry Gold, and Andrzej Ehrenfeucht. Information content of binding sites on nucleotide sequences. *Journal of Molecular Biology*, 188(3):415–431, April 1986. ISSN 0022-2836. doi: 10.1016/0022-2836(86)90165-8. URL http://dx.doi.org/10.1016/0022-2836(86)90165-8.

Eduardo S. Schwartz. The valuation of warrants: Implementing a new approach. *Journal of Financial Economics*, 4(1):79–93, January 1977. ISSN 0304-405X. doi: 10.1016/0304-405x(77)90037-x. URL http://dx.doi.org/10.1016/0304-405X(77)90037-X.

David Sherrington and Scott Kirkpatrick. Solvable model of a spin-glass. *Physical Review Letters*, 35(26):1792–1796, December 1975. ISSN 0031-9007. doi: 10.1103/physrevlett.35.1792. URL http://dx.doi.org/10.1103/PhysRevLett.35.1792.

W. Shockley. The theory of p-n junctions in semiconductors and p-n junction transistors. *Bell System Technical Journal*, 28(3):435–489, July 1949. ISSN 0005-8580. doi: 10.1002/j.1538-7305.1949.tb03645.x. URL http://dx.doi.org/10.1002/j.1538-7305.1949.tb03645.x.

Gilbert Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5(3):506–517, September 1968. ISSN 1095-7170. doi: 10.1137/0705041. URL http://dx.doi.org/10.1137/0705041.

J. Swift and P. C. Hohenberg. Hydrodynamic fluctuations at the convective instability. *Physical Review A*, 15(1):319–328, January 1977. ISSN 0556-2791. doi: 10.1103/physreva.15.319. URL http://dx.doi.org/10.1103/PhysRevA.15.319.

5 Team, Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, Kedong Wang, Lucen Zhong, Mingdao Liu, Rui Lu, Shulin Cao, Xiaohan Zhang, Xuancheng Huang, Yao Wei, Yean Cheng, Yifan An, Yilin Niu, Yuanhao Wen, Yushi Bai, Zhengxiao Du, Zihan Wang, Zilin Zhu, Bohan Zhang, Bosi Wen, Bowen Wu, Bowen Xu, Can Huang, Casey Zhao, Changpeng Cai, Chao Yu, Chen Li, Chendi Ge, Chenghua Huang, Chenhui Zhang, Chenxi Xu, Chenzheng Zhu, Chuang Li, Congfeng Yin, Daoyan Lin, Dayong Yang, Dazhi Jiang, Ding Ai, Erle Zhu, Fei Wang, Gengzheng Pan, Guo Wang, Hailong Sun, Haitao Li, Haiyang Li, Haiyi Hu, Hanyu Zhang, Hao Peng, Hao Tai, Haoke Zhang, Haoran Wang, Haoyu Yang, He Liu, He Zhao, Hongwei Liu, Hongxi Yan, Huan Liu, Huilong Chen, Ji Li, Jiajing Zhao, Jiamin Ren, Jian Jiao, Jiani Zhao, Jianyang Yan, Jiaqi Wang, Jiayi Gui, Jiayue Zhao, Jie Liu, Jijie Li, Jing Li, Jing Lu, Jingsen Wang, Jingwei Yuan, Jingxuan Li, Jingzhao Du, Jinhua Du, Jinxin Liu, Junkai Zhi, Junli Gao, Ke Wang, Lekang Yang, Liang Xu, Lin Fan, Lindong Wu, Lintao Ding, Lu Wang, Man Zhang, Minghao Li, Minghuan Xu, Mingming Zhao, Mingshu Zhai, Pengfan Du, Qian Dong, Shangde Lei, Shangqing Tu, Shangtong Yang, Shaoyou Lu, Shijie Li, Shuang Li, Shuang-Li, Shuxun Yang, Sibo Yi, Tianshu Yu, Wei Tian, Weihan Wang, Wenbo Yu, Weng Lam Tam, Wenjie Liang, Wentao Liu, Xiao Wang, Xiaohan Jia, Xiaotao Gu, Xiaoying Ling, Xin Wang, Xing Fan, Xingru Pan, Xinyuan Zhang, Xinze Zhang, Xiuqing Fu, Xunkai Zhang, Yabo Xu, Yandong Wu, Yida Lu, Yidong Wang, Yilin Zhou, Yiming Pan, Ying Zhang, Yingli Wang, Yingru Li, Yinpei Su, Yipeng Geng, Yitong Zhu, Yongkun Yang, Yuhang Li, Yuhao Wu, Yujiang Li, Yunan Liu, Yunqing Wang, Yuntao Li, Yuxuan Zhang, Zezhen Liu, Zhen Yang, Zhengda Zhou, Zhongpei Qiao, Zhuoer Feng, Zhuorui Liu, Zichen Zhang, Zihan Wang, Zijun Yao, Zikang Wang,

Ziqiang Liu, Ziwei Chai, Zixuan Li, Zuodong Zhao, Wenguang Chen, Jidong Zhai, Bin Xu, Minlie Huang, Hongning Wang, Juanzi Li, Yuxiao Dong, and Jie Tang. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models, 2025. URL https://arxiv.org/abs/2508.06471.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, S. H. Cai, Yuan Cao, Y. Charles, H. S. Che, Cheng Chen, Guanduo Chen, Huarong Chen, Jia Chen, Jiahao Chen, Jianlong Chen, Jun Chen, Kefan Chen, Liang Chen, Ruijue Chen, Xinhao Chen, Yanru Chen, Yanxu Chen, Yicun Chen, Yimin Chen, Yingjiang Chen, Yuankun Chen, Yujie Chen, Yutian Chen, Zhirong Chen, Ziwei Chen, Dazhi Cheng, Minghan Chu, Jialei Cui, Jiaqi Deng, Muxi Diao, Hao Ding, Mengfan Dong, Mengnan Dong, Yuxin Dong, Yuhao Dong, Angang Du, Chenzhuang Du, Dikang Du, Lingxiao Du, Yulun Du, Yu Fan, Shengjun Fang, Qiulin Feng, Yichen Feng, Garimugai Fu, Kelin Fu, Hongcheng Gao, Tong Gao, Yuyao Ge, Shangyi Geng, Chengyang Gong, Xiaochen Gong, Zhuoma Gongque, Qizheng Gu, Xinran Gu, Yicheng Gu, Longyu Guan, Yuanying Guo, Xiaoru Hao, Weiran He, Wenyang He, Yunjia He, Chao Hong, Hao Hu, Jiaxi Hu, Yangyang Hu, Zhenxing Hu, Ke Huang, Ruiyuan Huang, Weixiao Huang, Zhiqi Huang, Tao Jiang, Zhejun Jiang, Xinyi Jin, Yu Jing, Guokun Lai, Aidi Li, C. Li, Cheng Li, Fang Li, Guanghe Li, Guanyu Li, Haitao Li, Haoyang Li, Jia Li, Jingwei Li, Junxiong Li, Lincan Li, Mo Li, Weihong Li, Wentao Li, Xinhang Li, Xinhao Li, Yang Li, Yanhao Li, Yiwei Li, Yuxiao Li, Zhaowei Li, Zheming Li, Weilong Liao, Jiawei Lin, Xiaohan Lin, Zhishan Lin, Zichao Lin, Cheng Liu, Chenyu Liu, Hongzhang Liu, Liang Liu, Shaowei Liu, Shudong Liu, Shuran Liu, Tianwei Liu, Tianyu Liu, Weizhou Liu, Xiangyan Liu, Yangyang Liu, Yanming Liu, Yibo Liu, Yuanxin Liu, Yue Liu, Zhengying Liu, Zhongnuo Liu, Enzhe Lu, Haoyu Lu, Zhiyuan Lu, Junyu Luo, Tongxu Luo, Yashuo Luo, Long Ma, Yingwei Ma, Shaoguang Mao, Yuan Mei, Xin Men, Fanqing Meng, Zhiyong Meng, Yibo Miao, Minqing Ni, Kun Ouyang, Siyuan Pan, Bo Pang, Yuchao Qian, Ruoyu Qin, Zeyu Qin, Jiezhong Qiu, Bowen Qu, Zeyu Shang, Youbo Shao, Tianxiao Shen, Zhennan Shen, Juanfeng Shi, Lidong Shi, Shengyuan Shi, Feifan Song, Pengwei Song, Tianhui Song, Xiaoxi Song, Hongjin Su, Jianlin Su, Zhaochen Su, Lin Sui, Jinsong Sun, Junyao Sun, Tongyu Sun, Flood Sung, Yunpeng Tai, Chuning Tang, Heyi Tang, Xiaojuan Tang, Zhengyang Tang, Jiawen Tao, Shiyuan Teng, Chaoran Tian, Pengfei Tian, Ao Wang, Bowen Wang, Chensi Wang, Chuang Wang, Congcong Wang, Dingkun Wang, Dinglu Wang, Dongliang Wang, Feng Wang, Hailong Wang, Haiming Wang, Hengzhi Wang, Huaqing Wang, Hui Wang, Jiahao Wang, Jinhong Wang, Jiuzheng Wang, Kaixin Wang, Linian Wang, Qibin Wang, Shengjie Wang, Shuyi Wang, Si Wang, Wei Wang, Xiaochen Wang, Xinyuan Wang, Yao Wang, Yejie Wang, Yipu Wang, Yiqin Wang, Yucheng Wang, Yuzhi Wang, Zhaoji Wang, Zhaowei Wang, Zhengtao Wang, Zhexu Wang, Zihan Wang, Zizhe Wang, Chu Wei, Ming Wei, Chuan Wen, Zichen Wen, Chengjie Wu, Haoning Wu, Junyan Wu, Rucong Wu, Wenhao Wu, Yuefeng Wu, Yuhao Wu, Yuxin Wu, Zijian Wu, Chenjun Xiao, Jin Xie, Xiaotong Xie, Yuchong Xie, Yifei Xin, Bowei Xing, Boyu Xu, Jianfan Xu, Jing Xu, Jinjing Xu, L. H. Xu, Lin Xu, Suting Xu, Weixin Xu, Xinbo Xu, Xinran Xu, Yangchuan Xu, Yichang Xu, Yuemeng Xu, Zelai Xu, Ziyao Xu, Junjie Yan, Yuzi Yan, Guangyao Yang, Hao Yang, Junwei Yang, Kai Yang, Ningyuan Yang, Ruihan Yang, Xiaofei Yang, Xinlong Yang, Ying Yang, Yi Yang, Yi Yang, Zhen Yang, Zhilin Yang, Zonghan Yang, Haotian Yao, Dan Ye, Wenjie Ye, Zhuorui Ye, Bohong Yin, Chengzhen Yu, Longhui Yu, Tao Yu, Tianxiang Yu, Enming Yuan, Mengjie Yuan, Xiaokun Yuan, Yang Yue, Weihao Zeng, Dunyuan Zha, Haobing Zhan, Dehao Zhang, Hao Zhang, Jin Zhang, Puqi Zhang, Qiao Zhang, Rui Zhang, Xiaobin Zhang, Y. Zhang, Yadong Zhang, Yangkun Zhang, Yichi Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yushun Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Chenguang Zhao, Feifan Zhao, Jinxiang Zhao, Shuai Zhao, Xiangyu Zhao, Yikai Zhao, Zijia Zhao, Huabin Zheng, Ruihan Zheng, Shaojie Zheng, Tengyang Zheng, Junfeng Zhong, Longguang Zhong, Weiming Zhong, M. Zhou, Runjie Zhou, Xinyu Zhou, Zaida Zhou, Jinguo Zhu, Liya Zhu, Xinhao Zhu, Yuxuan Zhu, Zhen Zhu, Jingze Zhuang, Weiyu Zhuang, Ying Zou, and Xinxing Zu. Kimi k2.5: Visual agentic intelligence, 2026. URL https://arxiv.org/abs/2602.02276.

Minyang Tian, Luyu Gao, Shizhuo Dylan Zhang, Xinan Chen, Cunwei Fan, Xuefei Guo, Roland Haas, Pan Ji, Kittithat Krongchon, Yao Li, Shengyan Liu, Di Luo, Yutao Ma, Hao Tong, Kha Trinh, Chenyu Tian, Zihan Wang, Bohao Wu, Yanyu Xiong, Shengzhu Yin, Minhui Zhu, Kilian Lieret, Yanxin Lu, Genglin Liu, Yufeng Du, Tianhua Tao, Ofir Press, Jamie Callan, Eliu Huerta, and Hao Peng. Scicode: A research coding benchmark curated by scientists, 2024. URL https://arxiv.org/abs/2407.13168.

UK AI Security Institute. Inspect evals: SciCode. https://ukgovernmentbeis.github.io/inspect_evals/evals/coding/scicode/, 2026. Accessed 2026-07.

Marc D. Ulrich, William F. Seng, and Peter A. Barnes. Solutions to the fermi-dirac integrals in semiconductor physics using polylogarithms. *Journal of Computational Electronics*, 1(3):431–434, October 2002. ISSN 1572-8137. doi: 10.1023/a:1020784532229. URL http://dx.doi.org/10.1023/A:1020784532229.

Joshua Vendrow, Edward Vendrow, Sara Beery, and Aleksander Madry. Do large language model benchmarks test reliability?, 2025. URL https://arxiv.org/abs/2502.03461.

Sean Vig, Anshul Kogar, Matteo Mitrano, Ali Husain, Luc Venema, Mindy Rak, Vivek Mishra, Peter Johnson, Genda Gu, Eduardo Fradkin, Michael Norman, and Peter Abbamonte. Measurement of the dynamic charge response of materials using low-energy, momentum-resolved electron energy-loss spectroscopy (m-eels). *SciPost Physics*, 3(4), October 2017. ISSN 2542-4653. doi: 10.21468/scipostphys.3.4.026. URL http://dx.doi.org/10.21468/SciPostPhys.3.4.026.

Zihan Wang, Yu Fu, Akshit Goyal, and Sergei Maslov. Fitness advantage of sequential metabolic strategies emerges from community interactions in strongly fluctuating environments. *bioRxiv*, June 2024. doi: 10.1101/2024.06.14.599039. URL http://dx.doi.org/10.1101/2024.06.14.599039.

Steven R. White. Density matrix formulation for quantum renormalization groups. *Physical Review Letters*, 69(19):2863–2866, November 1992. ISSN 0031-9007. doi: 10.1103/physrevlett.69.2863. URL http://dx.doi.org/10.1103/PhysRevLett.69.2863.

B. Widom. Some topics in the theory of fluids. *The Journal of Chemical Physics*, 39(11):2808–2812, December 1963. ISSN 1089-7690. doi: 10.1063/1.1734110. URL http://dx.doi.org/10.1063/1.1734110.

Boxi Yu, Yuxuan Zhu, Pinjia He, and Daniel Kang. Utboost: Rigorous evaluation of coding agents on swe-bench, 2025. URL https://arxiv.org/abs/2506.09289.

Christopher Zanoli, Andrea Giovannini, Tengjun Jin, Ana Klimovic, and Yotam Perlitz. Elt-bench-verified: Benchmark quality issues underestimate ai agent capabilities, 2026. URL https://arxiv.org/abs/2603.29399.

Lifa Zhang and Qian Niu. Angular momentum of phonons and the einstein–de haas effect. *Physical Review Letters*, 112(8), February 2014. ISSN 1079-7114. doi: 10.1103/physrevlett.112.085503. URL http://dx.doi.org/10.1103/PhysRevLett.112.085503.

A AUDIT AND CORRECTION WORKFLOW

We went through the test problems one at a time. For each, we ran its reference solution against its own test cases, marked every defect we found—a wrong or non-reproducible gold, an over-tight tolerance, RNG-dependent grading, an unspecified convention, or a truncated or self-contradictory prompt—and applied a targeted fix, regenerating the affected gold targets when a text edit alone could not resolve it. Every change is logged, every corrected reference solution passes its own (corrected) tests before release, and the released dataset and evaluation files are rebuilt directly from these per-problem corrections.

The audit ran as successive review rounds, each structured as audit → fix → adversarial re-review → release gate. Each approved change is recorded (problem, field, before, after, rationale, round) and applied to a single per-problem source file, from which the released files are regenerated. Before each release, an automated check verifies both directions—every recorded change appears in the release, and every changed field is backed by a record—and re-runs the gold self-tests. Every fix then faced an independent adversarial re-review that had to reproduce the defect and attempt to refute the fix. Where it could not confirm, fixes were overturned and redone. Problem 13 is a documented example: a parity fix triggered a re-audit and a test redesign spanning six downstream subproblems.

Two disclosures. First, regenerating the gold targets changed some precomputed values: 69 target groups differ from the upstream release. 64 of these correspond to recorded changes; the remaining five (subproblems 1.1, 61.1, 69.5, 73.1, 80.6) changed only because byte-identical gold code yields different values in a modern software environment. On four of those five, all four models with matched before/after runs (§3) flip from fail to pass; the upstream precomputed values had become unreachable. These five are *not* counted among the 263 defects. Second, each problem was audited by one domain expert and every approved fix re-checked by a second. Because that second pass confirms rather than independently annotates, we report no inter-annotator agreement (§6).

B TWO-ENVIRONMENT OR GRADING: MEASURED EFFECT

Grading runs each step's tests under two pinned environments—`2024` (numpy 1.26/scipy 1.13 era) and `2025` (current)—and a step passes if it passes in either (§3.1). Table 5 reports, per model, the steps whose verdict the OR rule changes relative to the `2024` stack alone, extracted from the released per-step grading caches (`analysis/or_gain.json`).

Model	Steps rescued (of 287)	Step ids	Main problems flipped
GPT-5.5	8	65.6, 80.1–80.7	2 (65, 80)
Gemini 3.5 Flash	7	65.6, 80.1–80.6	0
Claude Opus 4.8	9	60.5, 65.5–65.6, 80.1–80.6	2 (60, 65)
GLM-5.2	8	65.5–65.6, 80.1–80.6	1 (65)

Table 5: Steps that pass only under the `2025` grading stack. The rescued sets are nearly identical across models, so the OR rule favors no model.

Every diagnosed rescue is an environment artifact, not a correctness difference. (i) The `2024` stack lacks `matplotlib`. Problem 80's shared dependency header does `from mpl_toolkits.mplot3d import Axes3D`, so every step of that problem fails at import in that stack regardless of the submitted code; this accounts for 6–7 of each model's rescues. (ii) In problem 65, scipy's `sqrtm` emits extended-precision (`complex256`) arrays for near-singular density matrices, which the older numpy's `linalg` rejects with a `TypeError`; this accounts for 1–2 rescues per model, identically across independently written solutions. (iii) One rescue (Claude Opus 4.8, step 60.5, a large Monte Carlo) is verified from the cache but was not diagnosed within our reproduction time budget; we report it as undiagnosed. Two caveats. First, the grader tries the `2024` stack first and short-circuits on a pass, so the caches exhibit only the `2024`-fail → `2025`-pass direction (re-running a passing step in the second stack could not change an OR verdict). Second, main-problem effects differ across models only because of independent failures elsewhere in the same problems: Gemini 3.5 Flash, for example, gains no main problem despite seven rescued steps.

C COMPLETE PER-PROBLEM DEFECT LOG

This appendix lists every defect in the audit, grouped by problem and derived from the released decision ledger and the original→corrected diff. Each entry gives the subproblem, defect *nature*: *subtype* (SCIENTIFIC=needs physics/math judgment to detect, SURFACE=corpus/code/text; subtypes are named as in Table 4), direction, and a one-line scientific reason for the correction. Direction labels correspond to the columns of Table 3: "too-strict/wrong" ↔ Strict/wrong, "neutral-cleanup" ↔ Neutral, "too-lenient (test tightened)" ↔ Weak-test$^{\uparrow}$. The ledger also lists changes made for downstream consistency with other fixes; these are marked "downstream change" and are not counted as defects. Counts: 263 defects across 64 of the 65 audited problems. Problem 2 is recorded below as an invalid, non-reproducible target and is excluded from the released benchmark.

PROBLEM 2 — GAUSSIAN_BEAM_FOCUS *Physics, Optics* (1 defect)

- **problem-level specification and target** *scientific: Invalid / non-reproducible target* · too-strict/wrong
  The specification does not determine a unique beam-focus answer, while the stored gold is not reproducible by any standard method. A correct implementation therefore has no uniquely verifiable target against which it can be graded. We record this as a benchmark defect but exclude Problem 2 from the corrected release.

PROBLEM 5 — LANCZOS *Mathematics, Numerical Linear Algebra* (1 defect)

Original source: (Lanczos, 1950)

- **5.1** *scientific: Non-discriminating test (physics)* · too-lenient (test tightened)
  The original starting vector b is palindromic and near-symmetric, so paired with a symmetric tridiagonal A it produces a degenerate Krylov subspace that a flawed Lanczos implementation can pass and that yields an ambiguous gold target. The corrected b breaks the symmetry, ensuring the test exercises a genuine non-degenerate Lanczos run.

PROBLEM 8 — SPATIAL_FILTERS_III *Physics, Optics* (4 defects)

Original source: (Goodman, 1996)

- **8.1 (problem_description_main + sub_steps[0].step_description_prompt)** *surface: Truncation / garble / dropped constant* · too-strict/wrong
  The problem description contains a sentence describing low-pass behavior ('pass only the central maximum of the diffraction pattern') copied from a sibling problem, directly contradicting this task's cross-shaped high-pass filter objective and misleading any model that reads the spec.
- **8.1 (sub_steps[0].function_header docstring)** *scientific: Unspecified convention* · too-strict/wrong
  The docstring omits two facts required to match the gold output: that T is a binary {0,1} mask in the zero-frequency-centered (fftshift) layout, and that filtered_image is the real part of the inverse transform—so a correct solution returning the magnitude or a non-centered mask would be incorrectly rejected.
- **8.1 (problem_io)** *surface: Typo / formatting* · neutral-cleanup
  The I/O block contains two spelling errors ('bandwidth', 'Ouput') and names the second output 'output_image' instead of 'filtered_image', creating an inconsistency with the function header; these are surface text corrections with no effect on grading.
- **8.1 (target / gold)** *scientific: Spec↔gold contradiction* · too-strict/wrong
  The original gold mask used an inclusive (>=) boundary, keeping the |kx|=bandwidth and |ky|=bandwidth cross-lines, while the prompt explicitly states 'the filter masks should not include the bandwidth frequency'; a correct solution using strict '>' exclusion failed on the ~0.45% of boundary pixels, so the gold was regenerated with strict '>' to match the stated spec.

PROBLEM 9 — WEIGHTED_JACOBI *Mathematics, Numerical Linear Algebra* (1 defect)

Original source: (Briggs et al., 2000)

- **9.1** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The spec's update formula divides by (a_ii * omega^-1) but omits the (1-omega) x_i^(k) damping term, making it correct only at omega=1 and producing wrong iterates for omega=2/3 and omega=0.5. The corrected formula is the standard damped Jacobi: x_i^(k+1) = (1-omega) x_i^(k) + omega * (b_i - sum_{j!=i} a_ij x_j^(k)) / a_ii.

PROBLEM 11 — GADC_ENTANGLEMENT *Physics, Quantum Information/Computing* (5 defects)

Original source: (Chen et al., 2024)

- **problem_description_main; step 11.6; step 11.7; step 11.11; step 11.12 (step_description_prompt)** *surface: Truncation / garble / dropped constant* · neutral-cleanup
  Dataset ingestion dropped tokens across five description fields, producing ungrammatical fragments ('Write a function with and functions', 'receiver function', 'measurement in .', missing '1.' in the one-particle sector label); the corrections restore grammatical prose without changing the task.
- **step 11.1 (ket); step 11.3 (tensor) function_header** *surface: Interface / return-contract mismatch* · neutral-cleanup
  The function headers 'def ket(dim)' and 'def tensor()' omit parameters their own docstrings describe ('args' for ket, '*args' for tensor), making the signatures uncallable as specified; tensor's output type was also mis-stated as '2d array' when the Kronecker product of vectors is an nd array.
- **step 11.4 (apply_channel); step 11.8 (syspermute); step 11.9 (partial_trace) function_header** *scientific: Unspecified convention* · neutral-cleanup
  The sys and perm arguments of apply_channel, syspermute, and partial_trace leave the subsystem-index base unspecified; without knowing whether indices are 0-based or 1-based, a correct implementation using the opposite convention would be graded wrong, so the correction pins the convention to 1-based throughout.
- **step 11.10 (step_background, von Neumann entropy)** *scientific: Wrong gold (method / sign / symmetry)* · neutral-cleanup
  The von Neumann entropy spec displays the formula S = -tr(rho log_2 rho) but the accompanying prose states "In denotes the (natural) matrix logarithm," directly contradicting the log_2 shown. An implementer following the prose would compute entropy in nats using the natural logarithm, whereas following the formula gives entropy in bits; the two differ by a factor of ln(2) ~= 0.693. The correction replaces the erroneous prose label with "log_2 denotes the base-2 matrix logarithm," making spec and formula consistent so the intended unit (bits) is unambiguous.
- **step 11.12 (step_background, post-measurement state)** *scientific: Wrong gold (method / sign / symmetry)* · neutral-cleanup
  The spec's formula for the post-measurement state writes the numerator as tr(Pi rho Pi)/p, where tr(Pi rho Pi) = p is a scalar equal to 1, so the expression collapses to the scalar 1 rather than a density matrix. The correct expression is Pi rho Pi / p, which projects the pre-measurement state onto the measurement outcome and renormalizes it. Any implementation that follows the stated formula produces a scalar, making downstream quantities such as coherent information undefined or incorrect.

PROBLEM 12 — SCHRODINGER_DFT_WITH_SCF *Chemistry, Quantum Chemistry* (4 defects)

Original source: (Hartree, 1928)

- **12 (problem_io)** *surface: Interface / return-contract mismatch* · neutral-cleanup
  The problem-level I/O docstring lists inputs only through "tolerance" and omits the "iteration" parameter entirely, even though the scf_routine function signature and its tests require an explicit maximum-iteration count. A reader implementing the function from the spec alone has no way to know this argument exists, its type (int), or its role as the SCF loop bound. The fix adds the missing "iteration: the maximum number of self-consistent field iterations; int" line, making the spec complete and consistent with the actual interface.
- **12.2 (Numerov test_cases)** *scientific: Unspecified convention* · too-strict/wrong
  Numerov returns a raw wavefunction whose amplitude scales linearly with the unspecified first-step seed, so comparing the raw array against a fixed-seed gold rejects every correct-shape solution whose seed differs;

changing the seed from -1e-10 to -1e-3 avoids blow-up near the r=0 Coulomb/centrifugal singularity, and normalizing both arrays by their L2 norm makes the test scale-invariant.

- **12.3 (compute_Schrod step_description_prompt)** *scientific: Unspecified convention* · neutral-cleanup
  The variable names 'u_at_0' and 'up_at_0', stated before the integration direction, read as boundary conditions at r=0 rather than as the IVP seed at the largest-radius starting point, causing models to integrate from the wrong end; reordering the sentence to bind the seed explicitly to the largest-radius start removes the ambiguity.
- **12.14 (scf_routine prompt + test_cases)** *scientific: Invalid / non-reproducible target* · too-strict/wrong
  The original SCF setup—mixing ratio 0.5, a flat Hartree initial guess (-2 + 2Z), only 10 iterations, and tolerance 1e-7—does not converge to a physical solution, so the gold (charge_density, total_energy) is non-converged; the fix uses mixing 0.3, a physically motivated warm-start 2Z(1-exp(-2r)), 60 iterations, and tolerance 1e-6, producing a converged gold that a correct solver can reproduce.

PROBLEM 13 — MAXWELL_EQUATION_SOLVER *Physics, Optics* (7 defects)

Original source: (Knapp et al., 2002)

- **13.9** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  A_z is a polar vector, so its mirror parities on the x=0 and y=0 inner faces are coupled to those of A_x and A_y; the configuration's sigma=(-1,-1,+1) symmetry requires A_z=(-,-,-), but the original gold stores A_z=(+,+,-), violating both the x-face and y-face constraints and making the target irreconcilable with a correct physics-based solution.
- **13.10** *scientific: Invalid / non-reproducible target* · too-strict/wrong
  The 13.10 update_fields target was computed from the 13.9 derivatives using the erroneous A_z=(+,+,-) parity, so the error propagates identically into the time-stepped result; only A_z changes (by exactly factor*dt*delta_A_z per test case) while the other six field components remain correct.
- **13.11** *scientific: RNG-dependent grading* · too-strict/wrong
  The original test grades the stepper by exact byte-reproduction of gold's specific ICN (iterated Crank-Nicolson) run on physically meaningless inputs (fields=(x,y,z,x,y,z,1)); because ICN iteration count, substep-to-dt mapping, and averaging form are all unspecified, any other valid stable scheme diverges from gold at ~1e-2, far above the grading tolerance, so correct solvers fail. The fix grades instead against a converged RK4 reference on divergence-free physical inputs within a tolerance matched to the scheme's own second-order convergence floor.
- **13.13** *scientific: RNG-dependent grading* · too-strict/wrong
  The integrate function composes the same under-specified stepper and was graded by byte-reproducing the constraint series from gold's exact ICN (iterated Crank-Nicolson) integration; any valid stable time-integration scheme diverges from gold at the 1e-2 to 1 level, making the test a scheme-fingerprint check rather than a physics check. The corrected test compares the final 7-field state to a converged RK4 reference (atol=2e-4) and asserts the constraint series is finite and bounded.
- **13.15** *scientific: RNG-dependent grading* · too-strict/wrong
  The top-level main function was graded by byte-reproducing the constraint series from gold's exact ICN (iterated Crank-Nicolson) integration at coarse grids (n in {10,20,40,52}), coupling correctness to the arbitrary stepping scheme; any valid integrator differs from gold at the 1e-2 level. The corrected test compares the constraint series from a single fine-grid run (n=64) against a converged RK4 reference with a tight tolerance (atol=1e-6) chosen from the measured div-E amplification floor (~1.4e-7).
- **13.15** *surface: Interface / return-contract mismatch* · too-strict/wrong
  The main docstring declares the return type as 'list of tuples' where each tuple holds (time, constraint value), but the function actually returns a flat 1-D ndarray of constraint values (the array produced by integrate); a solver following the docstring emits (time, value) pairs and fails on shape mismatch despite implementing the correct physics.
- **13.12** *scientific: RNG-dependent grading* · too-strict/wrong
  check_constraint computes a pure divergence operator (‖div E‖) with no scheme freedom, but its original test first ran the ICN (iterated Crank-Nicolson) stepper on an over-specified field configuration and then evaluated the operator, so the graded quantity inherited the stepper's exact-reproduction coupling. Any correct implementation of check_constraint that nonetheless uses a different (equally valid) field evolution would produce a numerically distinct ‖div E‖ and fail the assertion. The fix decouples the two concerns entirely: E

is set directly to a known div-free dipole field (expected ‖div E‖ ~ 0) and to a linear field E=(x,y,z) (expected div = 3), making the check a deterministic, scheme-independent operator test with atol=1e-9.

PROBLEM 14 — BROWNIAN_MOTION_IN_THE_OPTICAL_TWEEZER *Physics, Optics* (2 defects)

Original source: (Mannella, 2004)

- **14.2 (problem_io / output spec)** *scientific: Spec↔gold contradiction* · too-strict/wrong
  The output spec names the return value as eta (the ratio of computed to theoretical MSD), but the step function returns x_MSD (the raw MSD), with eta computed afterward in the harness; the mislabel could cause a solver to return the ratio and be graded wrong.
- **14.2 (sub_steps[1].test_cases and general_tests)** *scientific: RNG-dependent grading* · too-strict/wrong
  The test calls calculate_msd with Navg=4000 unseeded stochastic trajectories, making the eta ratio non-deterministic; a correct implementation can flakily fail the 0.95 < eta < 1.05 band across runs; adding np.random.seed(1) before each call makes the test reproducible without removing the statistical tolerance.

PROBLEM 15 — CRANK_NICOLSON_FOR_TIME_DEPENDENT_SCHRODINGER *Physics, Computational Physics* (2 defects)

Original source: (Crank & Nicolson, 1947)

- **problem_description_main + sub_steps[0] (step 15.1) step_description_prompt** *surface: Truncation / garble / dropped constant* · too-strict/wrong
  The spec wrote 'hbar = times 10^-34 Js' with the mantissa missing entirely, leaving the constant without a usable numerical value; a solver cannot reproduce the correct Crank-Nicolson trajectory without knowing hbar = 1.0545718e-34 Js.
- **sub_steps[0] (step 15.1) function_header docstring** *surface: Interface / return-contract mismatch* · too-strict/wrong
  The Crank-Nicolson A and B matrices for the time-dependent Schrodinger equation carry the imaginary unit i in their diagonal and off-diagonal coefficients, so labeling their elements as 'float' rather than 'complex' contradicts the actual output type and misdirects implementors.

PROBLEM 16 — DAVIDSON_METHOD *Mathematics, Numerical Linear Algebra* (2 defects)

Original source: (Davidson, 1975)

- **16.2 (and problem-level Inputs/Output block)** *scientific: Unspecified convention* · neutral-cleanup
  The original output spec said only 'computed eigenvalues', leaving unspecified both the count (num_eigenvalues lowest) and the required ordering (ascending), so correct implementations returning a different count or unsorted eigenvalues would be wrongly rejected by the test.
- **16.2** *scientific: Non-discriminating test (physics)* · too-strict/wrong
  With near-zero off-diagonal noise (0.0 to 0.0001), the matrix is nearly diagonal, causing 0/0 in the Davidson energy-correction denominator and trapping correct solvers; raising noise to 0.05 removes this degeneracy and also breaks a diagonal-sort shortcut that passes the degenerate case but fails genuine Davidson iterations.

PROBLEM 17 — LINEAR_TETRAHEDRON_METHOD *Physics, Condensed Matter Physics* (2 defects)

Original source: (Lehmann & Taut, 1972)

- **17.1** *scientific: Unspecified convention* · too-strict/wrong
  The original docstring left the sign of e_ij (eps_j - eps_i vs eps_i - eps_j), which input serves as eps_0, and

the exact key-name format (e.g. 'e01') all unspecified, yet the hidden test checks those dict keys and values exactly, so any correct solution using a different but internally consistent convention fails.

- **17.2** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The original step omitted the normalization convention (tetrahedron volume = BZ volume, prefactor 6) and supplied a physically wrong DOS gold with inconsistent interval ratios; the corrected gold follows the standard Blochl linear-tetrahedron formula with the explicit prefactor stipulated.

PROBLEM 18 — NURBS *Mathematics, Computational Mechanics* (3 defects)

Original source: (Hughes et al., 2005)

- **18.1 / 18.2 docstrings + problem_io (function_header + problem_io fields)** *scientific: Unspecified convention* · too-strict/wrong
  The 18.1 header mislabels xi as a 'knot index integer' (it is a float parameter coordinate) and i as a 'polynomial index' (it is a 1-based basis-function index), and declares the output as a 1D array when it is a scalar float; the 18.2 header has the same scalar-vs-array error. Additionally, the row-major weight flattening convention w[(i_1-1)*n_2 + (i_2-1)] is never stated, so a correct column-major implementation is graded wrong.
- **18.2 test_cases / general_tests (and precomputed target)** *scientific: Invalid / non-reproducible target* · too-strict/wrong
  The first test passes w=[0], a single-element zero vector, when 28 weights are required; a correct rational NURBS implementation raises IndexError or divide-by-zero and cannot even run. The precomputed target [0.5, 1.0] is a shape-(2) artifact of a buggy reference; the correct scalar value is 0.48387096774193544 (= 15/31 under row-major indexing).
- **18.2 step_background** *surface: Truncation / garble / dropped constant* · neutral-cleanup
  The step background presents only the 1D rational basis R_{i,n} but the step itself requires implementing the 2D tensor-product form R_{i_1,i_2} = (w N_{i_1} N_{i_2}) / sum_{a,b}(w N_a N_b); the governing formula for the quantity being coded is absent from the background.

PROBLEM 20 — PHONON_ANGULAR_MOMENTUM *Physics, Condensed Matter Physics* (1 defect)

Original source: (Zhang & Niu, 2014)

- **20.2 (phonon_angular_momentum)** *scientific: Unspecified convention* · neutral-cleanup
  The original output description 'mode decomposed phonon angular momentum' is ambiguous between the bare l_qv^alpha and the Bose-weighted per-mode summand (n0(omega_qv)+1/2)*l_qv^alpha that enters the total L^alpha; without the explicit formula, a correct solver computing the bare quantity would be graded wrong.

PROBLEM 21 — ABSORPTION_COEFFICIENT_FOR_ALLOY_GAALAS *Material Science, Semiconductor Materials* (3 defects)

Original source: (Adachi, 1985)

- **21.1 (problem_description_main) + 21.2 (step_description_prompt)** *surface: Truncation / garble / dropped constant* · too-strict/wrong
  The rendered spec dropped the numeric coefficients from the bandgap formula (showing only '+ x' instead of '1.424 + 1.247x') and omitted the mantissas from the physical constants (electron charge and reduced Planck constant), making the problem unsolvable without the missing values.
- **21.2 (function_header)** *surface: Interface / return-contract mismatch* · too-strict/wrong
  The function signature listed C as a required argument while the docstring stated 'Default is 1', a self-contradiction; additionally, the Returns field falsely labelled the output in m^-1 when the physical prefactor is absorbed into C, making alpha_eff an arbitrary-scale intermediate rather than a physical absorption coefficient.
- **21.2 (test_cases + eV-convention in step_description_prompt)** *scientific: Non-discriminating test (physics)* · too-strict/wrong
  Two of the three test targets were exactly zero (below-bandgap, no absorption) and the third was O(1e-27) due to SI-unit omega in the denominator, so all targets fell within np.allclose's default atol=1e-8 and a trivial

'return 0' passed the entire step; rescaling energies to eV and replacing the redundant below-gap point with above-gap cases forces a correct implementation of alpha proportional to sqrt(hbar*omega - Eg)/omega.

PROBLEM 22 — BEAM_TRANSLATION_REEXPANSION *Physics, Optics* (8 defects)

Original source: (Gumerov & Duraiswami, 2004)

- **22.2 (sub_steps[1].step_background)** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The recurrence for the rotation coefficient T_n^{nu m} ends with -2(Q31+iQ32) a_n^nu T_n^{nu m}, but the correct Wigner-rotation identity requires +2; the minus sign produces a non-unitary T (residual ~0.3—0.6) and the precomputed targets were built from it, so every physically correct implementation was graded wrong.
- **22.2 (sub_steps[1].step_background)** *scientific: Unspecified convention* · too-strict/wrong
  The base case T_n^{nu 0}(Q) = sqrt(4pi/(2n+1)) Y_n^{-nu} references Y_n^m without specifying the normalization convention; the Gumerov-Duraiswami convention (Condon-Shortley-free P_n^{|m|}) is required to reproduce the precomputed targets, and implementations using the standard scipy convention pass only 1 of 7 test cases.
- **22.2 (sub_steps[1].step_background)** *surface: Broken cross-reference* · neutral-cleanup
  The phrase 'consistent with those in .' contains an empty citation target where the reference was stripped, leaving the source of the a_n^m and b_n^m definitions unresolvable; the correction points explicitly to step 22.1.
- **22.2 (sub_steps[1].test_cases)** *scientific: Non-discriminating test (physics)* · too-lenient (test tightened)
  The original three test cases use only axis-aligned or special rotation matrices where distinct spherical-harmonic normalization conventions produce identical values, allowing at least four wrong convention variants to pass 2 of 3 cases; the four added generic ZYZ rotation cases discriminate all six known wrong variants.
- **22.3 (sub_steps[2].test_cases) / general_tests** *scientific: Non-discriminating test (physics)* · too-lenient (test tightened)
  All four original 22.3 test cases place the source point in the z=0 plane, where the sign and conjugate structure of the rotation coefficients is undetectable, and one case reduces to a degenerate boolean ==0 assertion; these tests accept implementations with the wrong rotation-coefficient sign/conjugate convention, so the correction replaces the boolean case with a numeric allclose check and adds an off-plane case.
- **22.3 (sub_steps[2].step_description_prompt)** *surface: Truncation / garble / dropped constant* · neutral-cleanup
  The prompt 'Write a code to calculate the reexpansion coeffcient with and .' contains two empty cross-references where citation targets were stripped and a typo ('coeffcient'), leaving the sentence grammatically broken and the referenced context unresolvable; the correction removes the dangling clause and fixes the spelling.
- **22.3 (sub_steps[2].step_background)** *surface: Broken cross-reference* · neutral-cleanup
  The phrase 'The rotation matrix Q with the definition in can be generated by the Rodrigues rotation formula' has an empty cross-reference target, so the definition of Q is unresolvable; the correction names step 22.2 explicitly.
- **problem_io / 22.3 function_header** *scientific: Unspecified convention* · neutral-cleanup
  The parameter description for B specifies only its shape (N_t+1, 2*N_t+1) without stating the index packing, leaving the mapping between array columns and order s ambiguous; the correction pins the convention as B[l, s+N_t] is the coefficient B_l^s in both the problem I/O block and the 22.3 docstring.

PROBLEM 23 — BLAHUT_ARIMOTO *Physics, Quantum Information/Computing* (3 defects)

Original source: (Arimoto, 1972)

- **23 (problem_description_main) and 23.3 (step_background)** *surface: Typo / formatting* · neutral-cleanup
  The exp() terms in the Blahut-Arimoto update-rule formula are missing their outer closing parenthesis in both

the main problem description and the step 23.3 background, making the expression syntactically malformed and unparseable as written.

- **23.3** *scientific: Over-tight tolerance* · too-strict/wrong
  The gold target for the [[0.8,0.5],[0.2,0.5]] channel was computed at tolerance e=1e-5, which leaves the algorithm at a non-converged iterate whose value (approx 0.073167) depends on whether the mutual information is computed in nats or bits; the corrected tolerance e=1e-8 converges to the true channel capacity (approx 0.073194), which is robust across both exp-base conventions.
- **23.3** *surface: Trivially broken test* · neutral-cleanup
  The test suite contained 7 entries with two exact duplicates: the three-input channel and the binary symmetric channel each appeared twice, adding redundant assertions with no additional coverage and inflating the apparent test count without testing any distinct case.

PROBLEM 24 — BURGERS_EQUATION *Mathematics, Computational Mechanics* (3 defects)

Original source: (Lax, 1954)

- **problem_io + 24.3 function_header** *scientific: Unspecified convention* · too-strict/wrong
  The problem-level spec listed spurious domain-end inputs a and b (the domain is fixed at [-pi/2, pi/2]) and described the output as a 2d (n_t-1)*(n_x-1) matrix, directly contradicting the step-level header's 1d final-time array; n_t was also ambiguous between grid points and intervals. The correction removes the phantom inputs, fixes the output to a 1d array of size n_x-1, and defines n_t as the number of temporal grid points giving n_t-1 intervals.
- **24.1 + 24.3 step_description_prompt** *surface: Truncation / garble / dropped constant* · neutral-cleanup
  Dataset-extraction errors truncated two prompt passages: 'array of length n-' lost the '1.' and swallowed the start of the next sentence, and 'from and .' dropped the referenced step numbers 24.1 and 24.2. The correction restores the missing text with no mathematical change.
- **24.1 test_cases** *scientific: Non-discriminating test (physics)* · too-strict/wrong
  Odd cell counts (9, 99, 29) place the central Gauss quadrature node exactly on the IC discontinuity at x=0, making the cell-averaged value depend on which branch is assigned at the jump point; the precomputed target assumed u(0)=+1 while the spec defines x<=0 as sin-1 giving -4/9, so any correct solver using that branch would fail. Even cell counts (10, 100, 30 cells, giving n_x=11/101/31) place the discontinuity on a cell boundary with no quadrature node at x=0, removing the branch ambiguity entirely.

PROBLEM 25 — CRM_IN_CHEMOSTAT *Biology, Ecology* (1 defect)

Original source: (MacArthur, 1970)

- **25.1** *surface: Interface / return-contract mismatch* · neutral-cleanup
  The SpeciesGrowth docstring omits 'spc' (current species abundance, 1D array of length N) from its Inputs list despite being the first function parameter, leaving one argument entirely undocumented; the correction adds the missing entry.

PROBLEM 26 — CRM_IN_SERIAL_DILUTION *Biology, Ecology* (1 defect)

Original source: (Bloxham et al., 2024)

- **26 (problem-level problem_io; describes the final-step signature SimulatedCycles, step 26.3)** *scientific: Unspecified convention* · too-strict/wrong
  The problem-level I/O spec lists parameters for a single continuous ODE integration (res_init, tf, dt) that the actual final function SimulatedCycles does not accept, and omits all serial-dilution parameters (Rs, SPC_THRES, T, D, N_cycles) that it requires, making the spec irreconcilable with the correct implementation.

PROBLEM 27 — DESIGN_TRADE_OFFS_FOR_HIGH_SPEED_PHOTODETECTORS *Material Science, Semiconductor Materials* (4 defects)

Original source: (Bowers & Burrus, 1987)

- **27.2** *surface: Truncation / garble / dropped constant* · too-strict/wrong
  The spec states the electron charge as 'x10^-19 C' with the leading mantissa 1.6 omitted, leaving a dimensionless and meaningless constant; correct solutions using the standard value q = 1.6x10^-19 C are graded wrong.
- **27.2** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The background formula gives C = epsilon/d, omitting the detector area A from the numerator; parallel-plate capacitance is C = epsilonA/d, and A is an explicit step input, so solutions using the correct formula produce values differing by A from the gold standard.
- **27.2** *scientific: Spec↔gold contradiction* · too-strict/wrong
  The background mislabels the device as a p-i-p diode and asserts V0 = 0, directly contradicting the step's given input V0 (applied voltage); solutions that correctly use the provided V0 in the depletion-width formula are penalized by a gold computed with V0 forced to zero.
- **27.3** *surface: Interface / return-contract mismatch* · too-strict/wrong
  The docstring declares xi and f_3dB as scalar float, but the test harness passes xi as a NumPy array (linspace over 50 points) and indexes the returned result; implementations that correctly vectorize are mismatched by the under-specified scalar-only contract.

PROBLEM 28 — GAUSSIAN_BEAM_INTENSITY *Physics, Optics* (8 defects)

Original source: (Kogelnik & Li, 1966)

- **problem_io; sub_steps[1](28.2).function_header; sub_steps[2](28.3).function_header** *scientific: Unspecified convention* · neutral-cleanup
  The original described z, L1, and s as relative distances 'from the lens' or 'from the source' in meters, but the gold geometry treats all three as absolute positions along the propagation axis at millimeter scale (waist at z = s, lens at z = L1); the relative-distance reading is inconsistent with the targets, so solutions built on it were misgraded.
- **sub_steps[0](28.1).step_background** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The background Gaussian-beam field expression contains two garbled errors contradicting standard Kogelnik-Li theory: the Gouy phase term "arctan(z/f)" is juxtaposed with the bracketed exponent with no subtraction operator, making it a multiplicative factor rather than a subtracted phase, and the beam-waist formula at the focal planes reads 2*sqrt(w0) instead of the correct sqrt(2)*w0. Any student implementation derived directly from this background would produce a field with the wrong phase dependence and incorrect waist value, causing a correct solution based on standard theory to be marked wrong against code or expected output that follows the garbled spec.
- **sub_steps[0](28.1).step_description_prompt** *scientific: Unspecified convention* · too-strict/wrong
  The spec directs a Fourier-domain propagation without pinning which variant; the gold matches only the Fresnel transfer-function H = exp(-i*pi*lambda*z*(fx^2+fy^2)), and an impulse-response implementation differs from it by roughly 17.5% at the second output point, so correct impulse-response solutions were wrongly rejected.
- **sub_steps[0](28.1).function_header** *scientific: Unspecified convention* · too-strict/wrong
  The original left the overall amplitude scale of the input field Gau unspecified, but the target is computed from the unit-peak form E(x,y,0) = exp(-(x^2+y^2)/w0^2); a correct solution using any other prefactor (e.g. including 1/w(z)) would be rejected by the test.
- **sub_steps[0](28.1).return_line** *surface: Interface / return-contract mismatch* · too-strict/wrong
  The scaffold's return statement names the propagated field 'Gau_pro' (lowercase p) while the function body and downstream steps use 'Gau_Pro' (uppercase P), causing a NameError at runtime for any submitted solution that follows the documented interface.
- **problem_io; sub_steps[2](28.3).function_header** *scientific: Unspecified convention* · too-strict/wrong

The original defined focus_depth only as 'new focus position through the lens,' leaving ambiguous whether it is the analytic continuous minimum or the discrete-grid argmin; the target uses z[np.argmin(Wz)], so a solution returning the analytic focus value would be rejected even when physically correct.

- **sub_steps[0](28.1) target (gold)** *scientific: Invalid / non-reproducible target* · too-strict/wrong
  The original 28.1 gold output P2 cannot be reproduced by the Fresnel transfer-function method, the angular-spectrum method, analytic propagation, or Fresnel impulse-response, and contradicts the 28.3 target (which a Fresnel-TF implementation passes); the gold was regenerated with the Fresnel-TF method to make the prompt, 28.1, and 28.3 self-consistent.
- **problem_io; sub_steps[2](28.3).function_header** *surface: Typo / formatting* · neutral-cleanup
  The Intensity output description in the problem_io spec and the step 28.3 function header both contain the misspelling "new fcous" instead of "new focus." The correction is a pure surface cleanup: the misspelling does not alter the meaning, does not affect any return-value contract, and has no consequence for grading—a model returning the correct 2D intensity array passes either way.

PROBLEM 30 — HELIUM_SLATER_JASTROW_WAVEFUNCTION *Chemistry, Quantum Chemistry* (2 defects)

Original source: (Jastrow, 1955)

- **30.3** *scientific: Over-tight tolerance* · too-strict/wrong
  The numerical-laplacian RMSE divides by delta^2, amplifying last-digit rounding in value(), so the precomputed target is a roundoff fingerprint of the reference implementation rather than a physically meaningful value; any faithful but independently coded wavefunction fails the exact allclose check even though its RMSE is scientifically correct.
- **30.3** *scientific: Unspecified convention* · neutral-cleanup
  The spec listed only psi, (gradient psi)/psi, and (laplacian psi)/psi as required methods, silently omitting (kinetic energy)/psi, which is part of the MultiplyWF interface contract; the missing method was also never tested, leaving an untested gap between the specified and required interface.

PROBLEM 31 — INDEPENDENT_COMPONENT_ANALYSIS *Physics, Computational Physics* (3 defects)

Original source: (Hyvärinen & Oja, 2000)

- **31.1** *scientific: Unspecified convention* · too-strict/wrong
  The docstring left the standard-deviation convention unspecified, so a correct ddof=1 implementation was graded wrong against ddof=0 gold, and was also inconsistent with the ddof=1 covariance (np.cov) used in steps 31.2 and 31.3. The fix pins the convention to sample SD (ddof=1) throughout.
- **31.2** *scientific: Unspecified convention* · too-strict/wrong
  A whitened matrix is defined only up to orthogonal rotation and sign, so comparing Z to a single precomputed matrix with np.allclose rejects all other correct whitenings. The test is replaced by a gauge-invariant check: identity covariance plus a least-squares row-space test confirming Z spans the same row space as the centered input.
- **31.3** *scientific: Unspecified convention* · too-strict/wrong
  ICA solutions are identified only up to permutation, sign, and scale, so pinning the output with np.allclose against one precomputed matrix rejects all other valid separations. Additionally, the two test inputs (3x8 and 3x10 matrices) are too short to be uniquely separable. The fix replaces exact comparison with a permutation- and sign-invariant |corr| >= 0.99 matcher and substitutes identifiable blind source separation mixtures (sawtooth/uniform/laplace and sine/uniform/laplace).

PROBLEM 32 — MULTIPARTICLE_DYNAMICS_IN_THE_OPTICAL_TWEEZER_ARRAY *Physics, Computational Physics* (3 defects)

Original source: (Liu et al., 2020)

- **32.1** *surface: Typo / formatting* · neutral-cleanup
  The F_xy display equation closes with a dangling right. (an unmatched LaTeX delimiter) instead of right], causing mis-rendering; every other analogous F_xy formula in the same background closes with right].
- **32.2** *scientific: Unspecified convention* · too-strict/wrong
  The spec omits the on-site stiffness formula k_i = alpha E_i^2 / w^2, the speed-of-light constant c, and whether h is an absolute or relative step; a correct solution using the intended relative step h*R_ij would fail the gold because an absolute-h interpretation produces catastrophically different numerical derivatives.
- **32.3** *scientific: Over-tight tolerance* · too-strict/wrong
  In the Gamma=0 (no decay) case the total occupation is conserved exactly, but a correct Runge-Kutta integration over 100000 steps accumulates floating-point drift exceeding 1e-6; the original threshold therefore rejects valid solvers, and loosening to 1e-3 still enforces conservation while admitting legitimate numerical integration error.

PROBLEM 33 — `PHASE_DIAGRAM_CHERN_HALDANE_MODEL_V1` *Physics, Condensed Matter Physics* (4 defects)

Original source: (Haldane, 1988)

- **33.2** *scientific: Unspecified convention* · too-strict/wrong
  The Haldane model has two bands with C_lower = -C_upper, so returning the Chern number without specifying which band makes the sign ambiguous and causes any correct upper-band implementation to fail the signed-integer test.
- **33.2** *scientific: Over-tight tolerance* · too-strict/wrong
  A finite Brillouin-zone grid computes the Berry-curvature sum as a near-integer float, not exactly an integer; comparing this raw float with np.allclose rejects physically correct solutions that quantize to the right integer but land slightly off due to grid discretization.
- **33.3** *scientific: Over-tight tolerance* · too-strict/wrong
  At a coarse BZ grid (2*pi/30 spacing), cells near phase boundaries are quantized differently by equally valid discretization conventions; requiring exact agreement of the entire phase-diagram matrix via cmp_tuple_or_list therefore rejects correct implementations that disagree only on boundary cells.
- **33.3** *scientific: Unspecified convention* · neutral-cleanup
  The docstring described the output matrix as sweeping (m/t2) and phi but never fixed which axis is row and which is column, leaving a transposed but otherwise correct implementation indistinguishable from the intended orientation.

PROBLEM 34 — `PN_DIODE_BAND_DIAGRAM` *Material Science, Semiconductor Materials* (4 defects)

Original source: (Shockley, 1949)

- **problem-level I/O block + step 34.3 function_header** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The parameter comments label N_a as the n-type concentration and N_d as the p-type concentration, reversing the standard acceptor/donor convention; the gold solution uses N_a for the p-type (acceptor) region and N_d for the n-type (donor) region, so any correct solution following standard semiconductor notation is graded wrong.
- **step 34.2 prompt** *surface: Truncation / garble / dropped constant* · too-strict/wrong
  The prompt renders the elementary charge as 'x 10^-19 C' with the mantissa 1.6 dropped, so a solver cannot determine the correct value (1.6e-19 C) from the problem text alone and any solution using the standard constant is penalized for a transcription error in the spec.
- **step 34.3 prompt** *scientific: Unspecified convention* · too-strict/wrong
  The prompt never specifies the spatial sampling grid, yet the test compares the full conduction-band array element-wise with np.allclose; choices such as np.arange vs linspace, step size, or endpoint handling all change the array length and values, making any grid-consistent but unspecified implementation fail the length-sensitive comparison.

- **step 34.3 function_header** *scientific: Unspecified convention* · too-strict/wrong
  The output is described only as 'the potential distribution', leaving both the sign convention (conduction-band value V_CB = -phi vs electrostatic potential phi) and the zero reference undefined; with an element-wise np.allclose comparison, a sign-consistent but differently-defined answer is incorrectly rejected.

PROBLEM 35 — QUANTUM_DOT_ABSORPTION_SPECTRUM *Chemistry, Quantum Chemistry* (4 defects)

Original source: (Harrison & Valavanis, 2016)

- **35.2** *surface: Truncation / garble / dropped constant* · too-strict/wrong
  The spec sentence 'the coefficients i,j,k are at least' was truncated mid-clause, leaving the floor value absent and the constraint unrecoverable; a model must guess whether coefficients are non-negative integers, natural numbers, or real values, making any well-defined solution ambiguous.
- **35.2** *surface: Interface / return-contract mismatch* · neutral-cleanup
  The docstring's Input section lists x, y, and z but omits N, the required integer argument that controls how many of the smallest quadratic combinations are returned and directly sets the output array size. Because N is undocumented, a solver reading the spec cannot determine the function's full interface: it may omit N from the call, pass the wrong value, or return an incorrectly sized array, all of which cause grading to fail even when the underlying computation is correct. The fix adds "N (int): The number of smallest quadratic combinations to return." to the Input list, completing the interface contract.
- **35.3** *scientific: Spec↔gold contradiction* · too-strict/wrong
  The prompt stated the function returns energy levels, but the actual computation converts energies to photon wavelengths in nm; a correct implementation returning wavelengths would fail a test expecting energy-level values, so the mismatch suppresses correct solutions.
- **35.3** *surface: Broken cross-reference* · neutral-cleanup
  The background instruction 'by using' was cut off before naming the function, leaving a dangling cross-reference; completing it with generate_quadratic_combinations() restores the intended call to the prior sub-step without adding implementation guidance.

PROBLEM 36 — QUASI_FERMI_LEVELS_OF_PHOTO_RESISTOR_OUT_OF_EQUILIBRIUM *Material Science, Semiconductor Materials* (1 defect)

Original source: (Ulrich et al., 2002)

- **problem_description_main + sub_steps[1] (step 36.2)** *surface: Truncation / garble / dropped constant* · too-strict/wrong
  The original spec truncates the electron charge to 'times 10^-19 C', dropping the mantissa 1.602 entirely; any model that correctly uses e = 1.602e-19 C cannot be graded against a problem statement that supplies an unusable, dimensionally incomplete constant.

PROBLEM 37 — RAY_OPTICS_SPHERICAL_ABERRATION *Physics, Optics* (4 defects)

Original source: (Jenkins & White, 1976)

- **37.1, 37.2, 37.3** *surface: Interface / return-contract mismatch* · neutral-cleanup
  The step descriptions list 'light wavelength' (and 'grid scaling factor' in 37.2) as function inputs, but no function signature accepts these arguments; wavelength enters only implicitly through the per-surface refractive indices. The phantom parameters mislead implementers about the function interface.
- **37 (main problem: 37.1, 37.2, 37.3, problem_io, general_tests)** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The original signatures carry a spurious third glass index n3, making surface 2 a crown-to-crown interface (no refraction) and collapsing the cemented doublet to a single crown element. All precomputed targets encode this degenerate single-element system; the fix drops n3, models the true air/n_crown/n_flint/air stack, and recomputes targets and the 37.3 monotonicity test for the real over-corrected doublet.

- **37.1** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The paraxial image-distance formula uses the incidence angle i instead of the refraction angle i', giving l' = r*i/u' + r. The physically correct paraxial relation requires the refracted angle i', and the corresponding closed-form expression in eq (2) is wrong in the same way; a solver following the original spec literally computes a wrong image distance that the gold rejects.
- **37.2** *scientific: Wrong gold (method / sign / symmetry)* · neutral-cleanup
  Step 37.2 computes the marginal (non-paraxial) ray trace for incidence angles much greater than 5 degrees, where the small-angle approximation fails, yet both the background sentence and the docstring summary call it 'paraxial'. The self-contradictory label obscures which regime the function models.

PROBLEM 39 — REFLECTION_SPECTRA_FOR_A_DISTRIBUTED_BRAGG_REFLECTOR *Material Science, Semiconductor Materials* (2 defects)

Original source: (Corzine et al., 1991)

- **39.2** *scientific: Unspecified convention* · too-strict/wrong
  The original spec's piecewise rule forces the real part of theta to pi when (A+D)/2 > 1, giving cos(theta) = -|(A+D)/2|, which directly contradicts the defining relation cos(theta) = (A+D)/2; the correction adopts the standard principal-branch arccos so both real and complex regimes satisfy the definition consistently.
- **39.1** *surface: Interface / return-contract mismatch* · neutral-cleanup
  The original spec stated the return value as a flat tuple (A, B, C, D) of scalar elements, but downstream steps consume a 2x2 numpy array; the correction aligns the stated output contract with the actual required shape M = [[A, B], [C, D]].

PROBLEM 40 — SPLITING_OPERATOR *Mathematics, Computational Mechanics* (3 defects)

Original source: (Strang, 1968)

- **40.3 (sub_steps[2])** *scientific: Invalid / non-reproducible target* · too-strict/wrong
  The original spec used a nominal dx = dt/CFL = 0.05 in the diffusion discretization, but the actual grid spacing on x = linspace(-1, 1, N) is 2/(N-1) ~= 0.04878, producing a ~5%-wrong diffusion operator; additionally, the time loop ran one step past T. The correction makes dx = x[1]-x[0] authoritative and integrates exactly round(T/dt) steps, regenerating the gold to match the self-consistent scheme.
- **40.2 (sub_steps[1])** *scientific: Unspecified convention* · too-strict/wrong
  The original prompt did not state which operator receives the two dt/2 half-steps versus the single full-step dt in the Strang splitting, yet the gold is only reproduced (to ~1e-16) by assigning the reaction term u^2 to the half-steps and the diffusion term alpha*u_xx to the full step; the alternative reading fails all targets, so any correct implementation of the other convention is wrongly graded wrong.
- **40 (problem_description_main)** *surface: Typo / formatting* · neutral-cleanup
  The problem description contained spelling errors ('opeator', 'spliting', 'Eurler') and the phrasing 'first order Strang splitting scheme' misattributes the first-order accuracy to the splitting itself rather than to the forward-Euler sub-steps; the correction fixes the typos and clarifies that the composite scheme is first order because each sub-step uses one forward-Euler update.

PROBLEM 41 — STRUCTURAL_STABILITY_IN_SERIAL_DILUTION *Biology, Ecology* (2 defects)

Original source: (Wang et al., 2024)

- **41.3** *surface: Broken cross-reference* · neutral-cleanup
  The step description instructs the solver to compute 'the determinant of M' without ever defining M, leaving the intended matrix ambiguous; replacing M with 'the normalized conversion matrix' identifies the correct operand.
- **41.3** *scientific: Invalid / non-reproducible target* · too-strict/wrong
  The third (R=4) test case carries an incorrect growth-rate matrix g, preference ranking pref, depletion order,

and niche-length vector t, so the precomputed gold target is wrong; the corrected instance supplies consistent, recomputed values and re-generates the target against them.

PROBLEM 42 — THE_THRESHOLD_CURRENT_FOR_MULTI_QUANTUM_WELL_LASERS *Material Science, Semiconductor Materials* (2 defects)

Original source: (McIlroy et al., 1985)

- **42.1, 42.2** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The step-42.1 output is labeled 'the gain coefficient G_th at threshold condition' and step-42.2 re-labels the same quantity as 'the modal gain', but the function computes and passes forward the single-well peak gain g_w; both misnomers make the spec internally inconsistent and physically incorrect.
- **42.3** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The background formula states eta*I_th = J_th*w*L, inserting a spurious injection-efficiency factor: eta already appears in the preceding relation eta*J_th = n_w*J_w, so the total threshold current is simply I_th = J_th*w*L, and the extra eta causes any correctly derived current to be marked wrong.

PROBLEM 43 — TWO_END_FIBER_LASER_GENERATOR *Physics, Optics* (3 defects)

Original source: (Kelson & Hardy, 1998)

- **43.1 and 43.3 (function_header docstrings)** *scientific: Wrong gold (method / sign / symmetry)* · neutral-cleanup
  In steps 43.1 and 43.3, the docstrings label gamma_s and gamma_p as "Gain coefficient," but these symbols denote the modal overlap (filling) factor—a dimensionless geometric quantity describing the spatial confinement of the optical mode, not a gain. The step-43.3 prompt body already uses "overlap factors" for the same variables, making the docstring internally contradictory. The relabeling to "Overlap (filling) factor" corrects the physical description without changing any computed quantity or grading target.
- **43.3 prompt + output specification (and problem_io / function_header return shape)** *scientific: Invalid / non-reproducible target* · too-strict/wrong
  The original prompt omitted the saturation-power formulas (P_ssat, P_psat), the BVP grid (100 equally spaced points on [0, L]), the initial guess, solver tolerance (1e-9), max_nodes (100000), the output-power definition Pout = Ps(L)*(1-R2), and the output array shape (100,) sampled via sol.sol(z); any correct solver using different conventions would mismatch the precomputed gold and be graded wrong.
- **43.3 test_cases / general_tests (final test)** *scientific: Non-discriminating test (physics)* · too-lenient (test tightened)
  The original test asserted only that the inversion profile is monotone (nz[0] > nz[mid]), ignoring the returned output power Pout entirely; a solution producing a physically wrong Pout would pass, so the assertion is strengthened to also require Pout > 50 W.

PROBLEM 45 — FINITE_DIFFERENCE_HEAT_EQUATION *Mathematics, Computational Mechanics* (6 defects)

Original source: (Courant et al., 1928)

- **problem_io; sub_steps[0](step 45.1).function_header; sub_steps[3](step 45.4).function_header** *scientific: Wrong gold (method / sign / symmetry)* · neutral-cleanup
  The spec names both material-temperature parameters 'T1' (a duplicate) and mislabels 'alpha2' as 'heat conductivity' rather than 'thermal diffusivity'; alpha2 is the diffusivity used throughout the update equations, so the wrong physical-quantity name contradicts the body text and misleads implementers.
- **sub_steps[0](step 45.1).step_description_prompt** *scientific: Unspecified convention* · too-strict/wrong
  The original 'second and third dimensions are x and y coordinates' is ambiguous about which axis is rows versus columns; the gold array shape (Nt, Ny, Nx) places y in the second dimension and x in the third, so the original wording can lead a correct solver to transpose the grid and fail all tests.

- **sub_steps[1](step 45.2).step_description_prompt; sub_steps[2](step 45.3).step_description_prompt** *surface: Broken cross-reference* · neutral-cleanup
  Both step 45.2 and 45.3 prompts contain a garbled cross-reference reading "defined in ." where the referent step number was dropped during authoring, leaving "the temperature array defined in ." with no resolvable antecedent. A model solving these steps cannot determine which prior step establishes the temperature array's structure, boundary layout, or indexing convention. The fix supplies the missing referent: "defined in step 45.1."
- **sub_steps[2](step 45.3).step_description_prompt** *scientific: Unspecified convention* · too-strict/wrong
  The original 'outward normal' sign convention for Neumann boundaries contradicts the gold implementation, which uses a positive-axis convention (T_b = T_in - N at the first row/column and T_b = T_in + N at the last); a solver following the outward-normal reading produces the wrong sign on three of four boundary faces.
- **sub_steps[1](step 45.2).test_cases; sub_steps[2](step 45.3).test_cases** *surface: Trivially broken test* · too-strict/wrong
  The test passes the boundary-condition array as a bare Python list rather than an np.array, so array-style indexing or dtype operations inside a correct implementation raise a TypeError; a fourth Neumann test case covering the last-column boundary was also absent, leaving the corrected positive-axis sign convention untested on the high-index side.
- **sub_steps[3](step 45.4).step_description_prompt** *scientific: Unspecified convention* · too-strict/wrong
  The original spec names only 'central difference' and the stability time-step, leaving the time-integration scheme (forward Euler), the diffusivity evaluation (pointwise, non-conservative form critical at the two-material interface), the grid spacing (dx = dy = 1), and the BC-application timing all unspecified; any of these choices can produce results that diverge from gold without the solver being physically wrong.

PROBLEM 46 — HELIUM_ATOM_VMC *Chemistry, Quantum Chemistry* (2 defects)

Original source: (McMillan, 1965)

- **46.3** *scientific: RNG-dependent grading* · too-strict/wrong
  The original test pins the exact RNG consumption order of a 2000-step seeded Metropolis walk via np.allclose against a precomputed trajectory, which rejects any correct implementation that uses a different but valid step size, batch size, or sweep order; the fix asserts instead that the time-average mean electron-nuclear distance satisfies |<r> - 3/(2*alpha)| < 0.02, the exact analytic value for the Slater trial wavefunction.
- **46.4** *scientific: RNG-dependent grading* · too-strict/wrong
  The original test uses cmp_tuple_or_list exact-reproduction plus seeded interval checks that implicitly mandate a specific protocol (run Metropolis once, evaluate only the final configs, use ddof=0 standard error, do not average the chain), rejecting any valid time-averaging VMC estimator; the fix replaces this with 3-sigma statistical bounds against the analytic kinetic energy alpha^2, electron-nuclear potential -4*alpha, and electron-electron potential 5*alpha/8.

PROBLEM 48 — MEELS_CONVERSION *Physics, Condensed Matter Physics* (4 defects)

Original source: (Vig et al., 2017)

- **problem_io + step 48.4 (step_description_prompt)** *scientific: Unspecified convention* · too-strict/wrong
  The spec names the output as chi"(omega) (the imaginary part of the density response), but the gold test expects the positive-omega slice to be positive, which is only true for -chi"; a correct chi" implementation would fail the sign check.
- **step 48.2 (function_header)** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The function-header docstring labels the output V_eff as having units of "inverse of square angstrom" (Angstrom^-2), but the Coulomb matrix element V_eff is physically dimensioned in square angstroms (Angstrom^2)—off by inversion. A correct implementation that returns values in Angstrom^2 cannot be verified against this spec, because the stated unit directly contradicts the actual physical quantity being computed.

- **step 48.4 (step_description_prompt)** *scientific: Unspecified convention* · too-strict/wrong
  The antisymmetrization step never specifies how to evaluate S(-omega) on a non-uniform, non-symmetric grid; the gold uses linear interpolation, so a reasonable alternative such as nearest-neighbor lookup produces a different numerical result and fails the allclose check.
- **step 48.4 (test_cases)** *surface: Trivially broken test* · too-strict/wrong
  The test applies a boolean numpy mask directly to the raw return value of chi_cal; when the function returns a Python list (as the docstring specifies), this raises a TypeError and a correct solution is rejected without wrapping the return in np.array first.

PROBLEM 50 — REPLICA_SYMMETRY_BREAKING *Physics, Condensed Matter Physics* (7 defects)

Original source: (Sherrington & Kirkpatrick, 1975)

- **problem_background_main / sub_steps[3](step 50.4).step_background** *scientific: Unspecified convention* · neutral-cleanup
  The original spec defined J_ij as i.i.d. N(0,1)/sqrt(N) without stating that J must be symmetric (J_ij = J_ji, required for the SK Hamiltonian to be well-defined) or that the diagonal is zero (no self-interaction); the correction makes the symmetric, zero-diagonal construction explicit.
- **sub_steps[0](step 50.1).step_description_prompt** *scientific: RNG-dependent grading* · too-strict/wrong
  The constraint permitting only np.random.randint and np.random.rand is an artificial RNG-protocol requirement that pins the exact random-number stream rather than constraining physics, so any correct Metropolis implementation using other valid random draws fails the test.
- **sub_steps[0](step 50.1).test_cases** *scientific: RNG-dependent grading* · too-strict/wrong
  Asserting np.allclose(spins, target) on the final spin configuration requires bitwise reproduction of the gold MC RNG trajectory, not thermalization to a correct equilibrium; the replacement tests the mean equilibrated energy per spin against a physically derived reference (-0.4207 +/- 3*0.0223), accepting any correct equilibration.
- **sub_steps[3](step 50.4).step_description_prompt** *scientific: RNG-dependent grading* · too-strict/wrong
  The constraint permitting only np.random.randn and np.random.choice pins the RNG stream so that seeded exact (RSB, mean, std) reproduction tests match bitwise; any correct implementation using other valid sampling functions fails, and the ddof=0 convention for the reported standard deviation was also left unspecified.
- **sub_steps[3](step 50.4).test_cases** *scientific: RNG-dependent grading* · too-strict/wrong
  Asserting exact reproduction of the (RSB, mean, std) triple via np.allclose passes only under bitwise RNG replication of the gold protocol; the replacement tests physically meaningful phase behaviour, checking that at high T=1.5 RSB is False with narrow overlaps and at low T=0.5 RSB is True with broad overlaps.
- **sub_steps[2](step 50.3).step_description_prompt / sub_steps[2](step 50.3).step_background** *scientific: Unspecified convention* · neutral-cleanup
  The prompt gave no testable return criterion for replica-symmetry breaking, leaving 'analyze the overall overlap distribution' entirely open-ended; the correction specifies the explicit threshold (std(overlaps) > 1.4/sqrt(N)) so solvers know exactly what Boolean to return.
- **sub_steps[2](step 50.3).test_cases** *scientific: RNG-dependent grading* · neutral-cleanup
  The test inputs for analyze_rsb were drawn with np.random.normal without a seed, so the overlap standard deviation could straddle the RSB threshold across runs, making the Boolean assertion non-deterministically flaky; adding fixed seeds makes the test inputs reproducible without constraining the solver.

PROBLEM 52 — SHOOTING_ALGO_H_ATOM *Physics, Computational Physics* (6 defects)

Original source: (Haule, n.d.)

- **52.1** *scientific: Non-discriminating test (physics)* · too-lenient (test tightened)
  All three original tests set u=0, making the second-derivative term (l(l+1)/r^2 - 2Z/r - En)*u identically zero regardless of the potential; a no-op implementation returning [y[1], 0] passes every case. Setting u0=1.0 in the first test makes the potential term nonzero and actually constrains the radial-Schrodinger physics.

- **52.2** *surface: Interface / return-contract mismatch* · too-strict/wrong
  The docstring declared the output ur as a scalar float and the input grid as 'linespace', but every test target is a shape-(100,) float64 array passed over a logspace grid; a solver following the documented type and grid name fails both type checks and np.allclose against the array target.
- **52.2, 52.3** *scientific: Invalid / non-reproducible target* · too-strict/wrong
  Hydrogen bound states require negative energy; the original positive (scattering) energies produce inward-integrated profiles dominated by integrator noise, so the precomputed targets are unreproducible across integrator settings. Replacing En with negative bound-state values and narrowing the radius ranges yields well-conditioned integrals with consistent, regenerable targets.
- **52.4** *surface: Broken cross-reference* · too-strict/wrong
  The prompt referenced Shoot(En, R, l) with three arguments, but the actual function defined in step 52.3 requires a fourth parameter y0; calling the documented signature raises a TypeError. The prompt also described l as a 'maximum' angular momentum, but FindBoundStates takes a single l value with the outer loop handled externally.
- **52.4** *scientific: Over-tight tolerance* · too-strict/wrong
  The shallowest bound-state energies from inward integration are sensitive to integrator step size at the level of ~1e-7, so the default np.allclose tolerance (rtol=1e-5, atol=0) can reject a numerically correct solver. Adding atol=1e-5 admits this integrator-noise scatter while still rejecting physically wrong energies.
- **52.4** *surface: Trivially broken test* · too-lenient (test tightened)
  Using .any() in the assertion means the check passes whenever the first bound state has l=0, regardless of its energy value; for example, Bnd[0]=(0, 42.0) satisfies the original test. Replacing .any() with .all() requires both the angular momentum and the energy to match the target, making the test non-vacuous.

PROBLEM 53 — STOCHASTIC_LOTKA_VOLTERRA *Biology, Ecology* (4 defects)

Original source: (Gillespie, 1977)

- **problem_io; sub_steps[2](step 53.3).function_header; sub_steps[3](step 53.4).function_header** *scientific: Unspecified convention* · neutral-cleanup
  The phrase 'rounded up to one decimal point' literally means ceiling, but the gold uses round-to-nearest at one decimal place; the ambiguity causes a correct implementation using round-to-nearest to be graded wrong.
- **sub_steps[0](step 53.1).step_background; sub_steps[0](step 53.1).step_description_prompt** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The original waiting-time PDF f=(1/a)exp(-dt/a) inverts the rate, giving mean a instead of the correct 1/a; additionally, omitting the sampling scale left solvers to guess whether to pass a or 1/a to NumPy's exponential, with the wrong choice shifting drawn times by roughly six orders of magnitude.
- **sub_steps[1](step 53.2).step_description_prompt** *scientific: Unspecified convention* · too-strict/wrong
  The prompt is silent on whether an event whose time reaches or crosses T is recorded; the test compares exact time arrays, so including versus excluding the boundary event changes the output, marking a correct simulation wrong if it applies a different but physically valid boundary convention.
- **sub_steps[3](step 53.4).test_cases** *scientific: Over-tight tolerance* · too-strict/wrong
  Exact float equality on a spectral period estimate rounded to one decimal place rejects correct solutions that land on an adjacent decimal bin, including cases where ceiling and round-to-nearest differ by up to 0.1; replacing with abs difference < 0.11 admits only genuine one-decimal ambiguity while still rejecting wrong periods.

PROBLEM 54 — SUPG *Mathematics, Computational Mechanics* (8 defects)

Original source: (Brooks & Hughes, 1982)

- **problem_description_main (+ sub_steps[2] step 54.3 step_description_prompt)** *scientific: Spec↔gold contradiction* · too-strict/wrong
  The spec's tau formula uses the element Peclet number P^e = |a|h/2kappa, but the gold assembly is consistent only with a fixed constant P = |a| = 200 (independent of h and kappa=1); models obeying the stated formula compute a wrong tau and fail. Additionally, the 54.3 prompt said to add both Nitsche and SUPG terms,

which double-counts the SUPG volume contribution already assembled in 54.2; the correction restricts 54.3 to Nitsche boundary terms only.

- **sub_steps[1] (step 54.2) step_description_prompt** *surface: Truncation / garble / dropped constant* · too-strict/wrong
  The 54.2 step prompt omits the load-bearing constants a=200, kappa=1, and the tau formula; because the harness feeds each step in isolation (without the main problem description), a model has no source for these values and cannot assemble the correct matrix, forcing failure on missing information rather than capability.
- **sub_steps[1] (step 54.2) step_background** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The 3-point Gauss-Legendre nodes are listed as +-3/5 = +-0.6, but the correct values are +-sqrt(3/5) ~ +-0.775; using the wrong nodes mis-integrates the degree-3 source term (12x^2 times a basis derivative), producing an RHS vector b that cannot match the gold.
- **problem_io + sub_steps[1] (54.2) / sub_steps[2] (54.3) / sub_steps[3] (54.4) function_header** *surface: Interface / return-contract mismatch* · too-strict/wrong
  The spec documents the solution array and RHS vector as 1-dimensional arrays of size N+1 and M, but the gold returns 2-dimensional column vectors of shape (N+1,1) and (M,1); a solver returning the documented 1d shape fails the shape check in the L2Error test.
- **sub_steps[0] (step 54.1) step_description_prompt** *scientific: Unspecified convention* · too-strict/wrong
  The basis-function prompt gives the piecewise formulas but never defines the node coordinates (x_j = (j-1)h, 1-based index j) or states that each function returns 0 outside its support interval; without these, the piecewise assembly and downstream index arithmetic are ambiguous and inconsistent with the gold's zero-outside behavior.
- **sub_steps[3] (step 54.4) step_description_prompt** *surface: Truncation / garble / dropped constant* · neutral-cleanup
  The 54.4 prompt reads 'using , and functions' with the dependency names dropped, leaving the instruction grammatically incomplete and uninterpretable; restoring the missing names (basis, assemble, stabilization) is a text-integrity fix with no change to difficulty.
- **sub_steps[3] (step 54.4) test_cases + general_tests (L2Error helper)** *surface: Trivially broken test* · too-strict/wrong
  The L2Error test helper loops over range(1,N), skipping the last element, and reconstructs u_h as sol[e]*basis(e,..,etype=1) + sol[e+1]*basis(e+1,..,etype=1), which uses wrong node offsets and applies etype=1 to both nodes; the correct reconstruction is sol[e-1]*basis(e,..,etype=2) + sol[e]*basis(e+1,..,etype=1) over range(1,N+1), so the original test measured error against a wrong approximate solution.
- **sub_steps[1] (step 54.2) step_background** *scientific: Unspecified convention* · too-strict/wrong
  The 54.2 background describes A as only the convection-diffusion bilinear form and b as only the int 12x^2 omega term, omitting the SUPG volume contributions (int tau a omega_,x (a u_h,x) dx for A and int tau a omega_,x (12x^2) dx for b) that the gold assembly includes; a model following the background alone assembles a wrong-per-gold matrix and right-hand side.

PROBLEM 55 — SWIFT_HOHENBERG *Physics, Condensed Matter Physics* (6 defects)

Original source: (Swift & Hohenberg, 1977)

- **55.1** *scientific: Unspecified convention* · too-strict/wrong
  The original spec describes the split-step scheme only qualitatively, omitting the explicit-Euler nonlinear update u_1 = u_N + dt(eps*u_N - u_N - u_N^3), the exact k-space propagator exp[dt(2k^2/q0^2 - k^4/q0^4)], the dx=1 wavenumber convention, the real-part projection, and the integer step count, so any legitimate alternative discretization fails the precomputed targets even if physically correct.
- **55.2** *scientific: Unspecified convention* · too-strict/wrong
  The spec names Kx, Ky, and Sk as returned arrays without specifying the meshgrid orientation (indexing='ij' vs 'xy' transposes the result) or that Sk = |fft2(u)|^2 with no normalization, fftshift-centered; a correct solver choosing the other convention fails element-wise comparison against the precomputed target.
- **55.3** *scientific: Unspecified convention* · too-strict/wrong
  The original peak-detection recipe leaves the bin count, range, peak-finding method, and proximity tolerance ('some reasonable tolerance threshold') unspecified, making the precomputed target unreproducible; the

correction pins N//2 equal-width bins over [0, max k_r], bin-center reporting, scipy.signal.find_peaks with a height threshold, and a 0.5*q0 proximity window.

- **55.3** *surface: Broken cross-reference* · neutral-cleanup
  The spec lists a 'narrowness' criterion for peak_found_near_q0 that the actual detection algorithm (proximity + height only) never applies, and uses the inconsistent output name peak_near_q0_location instead of peak_location_near_q0; the correction removes the phantom criterion and aligns the identifier.
- **55.4** *surface: Broken cross-reference* · neutral-cleanup
  The 'set to 0 if no stripe is formed' clause was attached to the min_height input parameter, where it is meaningless (min_height is a caller-supplied threshold, not a return value); the correction moves it to the stripe_mode output, where it correctly describes the no-pattern return convention.
- **55.4** *scientific: Over-tight tolerance* · too-strict/wrong
  The test compared the full output tuple including the chaotic, RNG- and discretization-sensitive u field and raw Sk to precomputed bit-exact values; any correct solver using a legitimate scheme variant fails even when the physically meaningful observables (if_form_stripes and stripe_mode) match, so the correction replaces full-tuple comparison with shape checks on u/Sk plus allclose on stripe_mode and equality on if_form_stripes.

PROBLEM 57 — 1D_HARMONIC_OSCILLATOR_NUMEROV_SHOOTING *Physics, Computational Physics* (3 defects)

Original source: (Noumerov, 1924)

- **57.2** *scientific: Unspecified convention* · too-strict/wrong
  The original spec omits the Numerov seed values (u0 = u_b, u1 = u_b + step*up_b) and does not say whether step carries its sign, so any implementation using a different seed or abs(step) is rejected by the gold tests despite being a valid Numerov integrator.
- **57.4** *scientific: Unspecified convention* · too-strict/wrong
  The original spec says to count sign changes between consecutive elements but leaves zero-element handling undefined; the gold uses a product < 0 rule (zeros are sign-neutral), so an implementation treating a zero as a sign-flip returns the wrong node count and fails the test.
- **57.5** *scientific: Unspecified convention* · too-strict/wrong
  The spec does not state that bound-state energies are taken as the first trial energy where the node count changes, with no further refinement; the gold stores coarse first-crossing values on the energy grid (e.g. n=0 energy ~1.0289 rather than the exact eigenvalue 1.0), so a more accurate bisection-refined result is graded wrong.

PROBLEM 58 — TOLMAN_OPPENHEIMER_VOLKOFF_STAR *Physics, Astrophysics* (6 defects)

Original source: (Oppenheimer & Volkoff, 1939)

- **58.2** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The docstring names the return value 'eps: the specific internal energy' but the function eos_rho_from_press computes and returns density, so a docstring-following solver returns the wrong physical quantity and fails every test.
- **58.3** *surface: Broken cross-reference* · too-strict/wrong
  Step 58.3 describes eos_eps_from_press but tells the solver to take density rho as input (it takes pressure) and invokes the wrong function eos_rho_from_press in every test, making a correct implementation of the intended function untestable.
- **58.4** *surface: Broken cross-reference* · neutral-cleanup
  The step 58.4 prompt refers to helpers eps_from_press and rho_from_press, which are undefined; the functions established in prior steps are eos_eps_from_press and eos_rho_from_press, so the broken names prevent a solver from locating the required helpers.
- **58.4** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The background states the mass-continuity ODE as dm/dr = 4 pi r^3 mu, but the correct GR relation is dm/dr = 4 pi r^2 mu; the erroneous r^3 exponent contradicts the gold integrand and marks a physically correct implementation wrong.

- **58.5** *scientific: Unspecified convention* · too-strict/wrong
  The original prompt omits the conventions required to reproduce gold: the uniform radial grid from r=0 to r_max, the reference gauge phi(0)=0, the surface rule (first radius where P<=0), and the lapse formula sqrt(1-2M/R)*exp(-phi(R)); without these a correct-method solver cannot match the target output.
- **58.5** *scientific: Over-tight tolerance* · too-strict/wrong
  Default-tolerance np.allclose effectively demands exact agreement with one integrator's trajectory, rejecting correct solutions from other ODE solvers; additionally, the first test's grid (npoints=200, rmax=100000) is too coarse and numerically unstable to yield a well-resolved target, so both the tolerance and grid parameters suppress correct implementations.

PROBLEM 59 — VQE *Physics, Quantum Information/Computing* (3 defects)

Original source: (O'Malley et al., 2016)

- **59.1** *surface: Interface / return-contract mismatch* · too-strict/wrong
  The scaffold's return statement references 'Rz', a z-specific variable name never defined for the general-axis function 'rotation_matrices(axis, theta)', which builds 'R'; any solution following the documented output variable raises NameError or returns nothing.
- **59.2** *scientific: Non-discriminating test (physics)* · too-strict/wrong
  The test compares the complex inner product vdot(ansatz_o, ansatz_c) directly against the real product of norms; a physically valid ansatz differing by a global phase e^(i*phi) satisfies the Cauchy-Schwarz equality in magnitude but not in real/imaginary parts, so np.isclose fails and the correct state is rejected. The fix uses np.abs on the inner product for phase invariance and adds an explicit unit-norm check.
- **59.3** *scientific: Non-discriminating test (physics)* · neutral-cleanup
  The original third test case (U = CNOT21 * (H S_dag x H S_dag), psi = |1,-1j> x |1,-1j>/2) gives a Z1 expectation coinciding with earlier cases and adds no new coverage; it is replaced with U = I, psi = |10>, which exercises Z1 = -1 (first qubit in state |1>) as a distinct branch. A correct implementation passes both forms, so this is a test-coverage redesign with no effect on pass/fail direction.

PROBLEM 60 — WIDOM_PARTICLE_INSERTION *Material Science, Molecular Modeling* (7 defects)

Original source: (Widom, 1963)

- **60.2** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The spec labels the potential 'truncated and shifted' (V^{tr-sh}) but the case formula applies no shift: it returns V_LJ(r) for r < r_c and 0 otherwise. A true shifted potential subtracts V_LJ(r_c) from every pair, changing the insertion energy; the contradictory label misleads solvers into implementing the wrong variant.
- **60.3** *scientific: RNG-dependent grading* · too-strict/wrong
  The original tests assert exact float values for a single random insertion (e.g. 1.0185805629757558), which only pass if the solver reproduces the identical legacy-RNG call protocol. The correction checks the statistically correct ensemble observable mean(exp(-beta*dU)) over 20000 insertions with tolerance 0.02, accepting any correct RNG implementation.
- **60.3** *surface: Typo / formatting* · neutral-cleanup
  The original return description vaguely named 'the test particle' without specifying that exactly one particle is inserted at a uniformly random position per call; the correction makes the single-insertion contract explicit. This is a wording clarification with no effect on difficulty.
- **60.4** *scientific: Unspecified convention* · too-strict/wrong
  The original spec described only 'a regular grid ... properly positioned' without pinning cell-center offset, grid rounding (n = ceil(N^(1/3))), enumeration order (k fastest, then j, then i), or that the full n^3 sites are returned even when n^3 > N. Any of these ambiguities cause a correct but differently-gridded solver to fail the reference tests.
- **60.5** *surface: Interface / return-contract mismatch* · too-strict/wrong
  The prompt listed 'positions' and 'L' as parameters of the MC function, but the actual signature takes N and rho; neither 'positions' nor 'L' exists as an argument. A solver following the prose would construct the wrong call, failing before any physics is evaluated.

- **60.5** *scientific: Unspecified convention* · too-strict/wrong
  The return spec described 'corrected energy' and 'extended chemical potential' without giving the LJ long-range tail formulas (u_tail and mu_tail). Without the explicit (8/3)*pi*rho*epsilon*sigma^3*((1/3)*(sigma/r_c)^9 - (sigma/r_c)^3) expressions, a correct but differently-convention solver cannot match the reference mu_ext values the test checks.
- **60.5** *scientific: Invalid / non-reproducible target* · too-strict/wrong
  The test included reference mu_ext values for rho = 0.8 (16.14) and rho = 0.9 (54.56) produced by a short MC run that has not converged at those high densities; a correct implementation's stochastic estimate for these points falls outside the 0.1 mean-relative-error gate, causing spurious failures. Dropping the two unconverged density points fixes this.

PROBLEM 61 — XRAY_CONVERSION_I *Physics, Condensed Matter Physics* (3 defects)

Original source: (Busing & Levy, 1967)

- **61.3** *scientific: Wrong gold (method / sign / symmetry)* · neutral-cleanup
  The docstring for parameter z_s in step 61.3 labels it as a step size in the phi rotation, but the function implements a theta-rotation step, and the rest of the problem (including step 61.5) consistently uses z_s as a theta step. The mislabeled axis creates a direct contradiction between the documented parameter semantics and the actual rotation convention, which can cause implementers to parameterize or interpret the rotation incorrectly. Correcting phi to theta restores internal consistency with no change to task difficulty.
- **61.5** *surface: Broken cross-reference* · neutral-cleanup
  Step 61.5's background instructions contained two unresolved cross-references ('Employ step to calculate' and 'G matrix obtained from step') with no step number supplied, and dropped the leading list item '1.', leaving the two-step procedure unnavigable; corrected by naming step 2 (q_cal) and identifying G as the goniometer rotation matrix.
- **61.5** *surface: Truncation / garble / dropped constant* · neutral-cleanup
  In the step 61.5 function header, the docstring parameter description for z_s originally read "step size in the heta rotation": the backslash-t of "theta" was consumed as a tab escape, leaving a run of whitespace followed by the fragment "heta". The rotation axis is thus unnamed and the parameter description is unreadable. The fix restores the intended token to a legible "theta", a neutral textual correction that does not alter the task or grading difficulty.

PROBLEM 62 — DMRG *Physics, Condensed Matter Physics* (4 defects)

Original source: (White, 1992)

- **62.1** *surface: Interface / return-contract mismatch* · too-strict/wrong
  The scaffold defined only 'class EnlargedBlock' with its '__init__' and 'print_all' methods duplicated, while 'class Block' (required for single-site blocks) was entirely absent, making the provided header syntactically and logically unusable.
- **62.3** *surface: Broken cross-reference* · neutral-cleanup
  Three cross-references in the problem text were left blank ("from step ,", "defined in step", "as described in step ."), forcing solvers to guess which prior steps define the required operators and Hamiltonian; the correction fills in the correct step numbers (62.2, 62.4).
- **62.5** *scientific: Non-discriminating test (physics)* · too-strict/wrong
  The DMRG truncation step operates on a degenerate SU(2) multiplet, so the eigenbasis within the degenerate subspace is defined only up to an arbitrary orthogonal rotation; comparing raw operator-matrix entries rejects correct implementations that happen to use a different valid gauge. The fix checks basis-invariant quantities (eigenvalues of H, singular values of conn_Sz and conn_Sp) instead.
- **62.6** *surface: Trivially broken test* · neutral-cleanup
  The test suite for step 62.6 listed the run_dmrg(block, 100, 100, model_d) call twice—once as case 3 and again as a trailing case 6—producing a redundant duplicate entry with no distinct parameter variation. The corrected suite retains five unique test cases (varying m0 and m) and drops the repeated case, eliminating unnecessary redundancy without changing which solutions pass or fail.

PROBLEM 63 — ESTIMATING_STOCK_OPTION_PRICE *Mathematics, Computational Finance* (7 defects)

Original source: (Schwartz, 1977)

- **63.2** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The background equation gives the expiry payoff as V(t_max, p) = max(K-s, 0), which is the European put payoff, but the problem prices a European call whose correct terminal condition is V(t_max, p) = max(s-K, 0). This sign reversal in the boundary condition propagates through any backward-in-time PDE solver, producing option values that correspond to a put rather than a call. A correct implementation using the call payoff max(s-K, 0) cannot reproduce the wrong boundary specified in the background, so any such solution is graded as incorrect.
- **63.3** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The explicit finite-difference update formula writes the c-coefficient term as c*V^n_{j-1}, duplicating the left-neighbour index; the standard three-point central-difference stencil requires c*V^n_{j+1}. With the right neighbour absent the recurrence V^{n+1}_j = a*V^n_{j-1} + b*V^n_j + c*V^n_{j-1} is self-contradictory with the stencil stated just above it. Any reference solution or gold target generated from this formula propagates the wrong value at every time step, so a correct implementation of the intended scheme produces different numerical output and fails the comparison.
- **63.4** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The docstring labels the tridiagonal matrix D as shape (N_t-2)x(N_t-2), but D operates on interior price nodes, making its correct shape (N_p-2)x(N_p-2); the wrong dimension label contradicts step 63.3's own description and misdirects any solver implementation.
- **63.1** *scientific: Unspecified convention* · too-strict/wrong
  The output description says only 'np.linspace between p_min and p_max' without specifying that the grid is in log-price or that the bounds must stay in the given (descending) order; the test passes bounds already swapped, so an ascending linspace produces a different array and fails a correct implementation.
- **63.6** *scientific: Non-discriminating test (physics)* · too-strict/wrong
  The test sets S0=100 with min_price=200 (= strike/5), placing the initial stock price below the grid's lower bound; interpolating V at log(S0) over a log-price grid that does not span that point is undefined, making the test ill-posed. The fix moves S0=400 in-range and specifies the floor time-index and log-grid linear-interpolation convention.
- **63.3** *surface: Interface / return-contract mismatch* · neutral-cleanup
  The original docstrings for steps 63.3 and 63.4 do not state that D is a scipy.sparse matrix, leaving the cross-step type contract implicit; since the gold implementation returns and consumes a sparse matrix, the omission can cause silent type mismatches in downstream steps.
- **63.2** *scientific: Unspecified convention* · too-strict/wrong
  The original prompt ('correctly puts boundary condition ... 2D array w.r.t. time and price') does not specify which axis corresponds to price versus time, the log-price form s=e^p used in the payoff and boundary rows, or the corner-precedence order; a correct solution with a transposed layout or different corner assignment would be wrongly rejected by the deterministic test.

PROBLEM 64 — GCMC *Material Science, Molecular Modeling* (5 defects)

Original source: (Adams, 1975)

- **64.2** *surface: Trivially broken test* · too-strict/wrong
  The dist() function returns a single float, but the test indexed it with [0]; a correct scalar return cannot be subscripted this way, so the assertion would crash or silently mis-compare against a correct solution.
- **64.4** *surface: Interface / return-contract mismatch* · neutral-cleanup
  The spec listed a phantom argument 'integer i' absent from the function signature, omitted the required box length L, and misspelled 'posistions'; the corrected description matches the actual E_i(r, positions, L, sigma, epsilon) interface.
- **64.4** *scientific: Unspecified convention* · too-strict/wrong
  When two particles coincide under the minimum-image convention, the Lennard-Jones energy diverges; different correct implementations may return +inf or nan, so the original unspecified convention could cause a correct solution to fail the gold comparison.

- **64.6** *scientific: Spec↔gold contradiction* · too-strict/wrong
  The spec gave real SI argon constants (mass in kg, sigma in meters) alongside an incoherent instruction to 'make J and s dimensionless', while the test cases run in Lennard-Jones reduced units (epsilon = sigma = m = 1, k_B = 1); the contradictory spec forced incorrect dimensional choices that the gold rejects.
- **64.6** *scientific: Over-tight tolerance* · too-strict/wrong
  A finite Monte Carlo run (1e5 steps) produces an equilibrium particle-count average with inherent stochastic variance; a 1% relative tolerance is too tight to reliably accept a correct GCMC implementation, whereas 5% accommodates statistical fluctuations while still rejecting incorrect results.

PROBLEM 65 — GHZ_PROTOCOL_FIDELITY *Physics, Quantum Information/Computing* (6 defects)

Original source: (Bennett et al., 1996)

- **65.1** *surface: Interface / return-contract mismatch* · too-strict/wrong
  The function signature 'def tensor()' accepts no arguments, so any call passing matrices raises a TypeError; adding *args makes the signature match the docstring's own requirement of 'any number of nd arrays'.
- **65.1** *surface: Interface / return-contract mismatch* · too-strict/wrong
  The spec declares the Kronecker product always returns a 2-d array of floats, but tensoring two 1-d vectors yields a 1-d result; a correct implementation returning the right shape fails the shape check, and the dtype restriction wrongly rejects complex outputs.
- **65.2, 65.3, 65.6** *surface: Interface / return-contract mismatch* · too-strict/wrong
  Quantum-channel Kraus operators are generically complex-valued; labeling them 'array of floats' in steps 65.2, 65.3, and 65.6 mislabels the type and causes type-based checks to reject correct complex implementations.
- **65.2** *scientific: Unspecified convention* · too-strict/wrong
  The 'sys' parameter is described only as 'list of subsystems' with no indexing convention stated; a solver choosing 0-based indices targets the wrong subsystem, producing a physically different result that fails gold tests.
- **65.3, 65.6** *surface: Broken cross-reference* · neutral-cleanup
  Both 65.3 and 65.6 prompts contain truncated cross-references ('apply_channel function in .' and 'the protocol in given by') with the step number missing, leaving solvers without the pointer to the required prerequisite function.
- **65.4** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The original formula rho" = V rho' V^dagger is dimensionally impossible for a two-party state: V is 2^n x 2, so the two-party unitary must be VV and the adjoint acts on the left; the correct formula is rho" = (VV)^dagger rho' (VV), and any correct implementation of this is graded wrong against the broken spec.

PROBLEM 66 — KOLMOGOROV_CRESPI_POTENTIAL *Material Science, Molecular Modeling* (5 defects)

Original source: (Kolmogorov & Crespi, 2005)

- **problem_io + sub_steps[5] (step 66.6) step_description_prompt + sub_steps[5] (step 66.6) function_header** *scientific: Unspecified convention* · too-strict/wrong
  The spec states only 'KC potential energy' with no normalization, yet the test targets are per-atom averages (total double sum divided by Ntop + Nbot); a solver returning the raw double sum is wrongly rejected.
- **sub_steps[5] (step 66.6) step_description_prompt** *scientific: Spec↔gold contradiction* · too-strict/wrong
  The prompt lists KC parameter values (e.g. z0=3.416084, C0=20.021583) that contradict the function-header defaults (z0=3.370060885645178, C0=21.78333851687074) used to compute the test targets, so a solver following the prompt's values is incorrectly rejected; additionally, 'C = E-2' is a garbled literal.
- **sub_steps[1] (step 66.2) step_description_prompt + sub_steps[1] (step 66.2) function_header** *surface: Truncation / garble / dropped constant* · too-strict/wrong
  The prompt and header declare the normal-vector return shape as (natoms,) instead of (natoms, 3), causing solvers that build the correct 3-D array to be rejected, and the instruction for correcting sign is truncated mid-token ('multiplying by -' instead of 'multiplying by -1').

- **sub_steps[5] (step 66.6) test_cases + general_tests** *surface: Trivially broken test* · too-lenient (test tightened)
  The misplaced parenthesis 'np.abs(energy - energy_ref < 2)' first evaluates the comparison to a bool then takes its absolute value, making the tolerance check one-sided (only requiring energy < energy_ref + 2) and allowing arbitrarily large negative errors to pass; the corrected form 'np.abs(energy - energy_ref) < 2' enforces the intended two-sided bound.
- **sub_steps[0] (step 66.1) step_description_prompt + sub_steps[0] (step 66.1) function_header** *scientific: Unspecified convention* · too-strict/wrong
  The spec leaves the graphene geometry construction entirely unspecified (no lattice vectors, no basis-atom ordering, no loop convention), yet the tests use np.allclose on absolute atom coordinates, so any correct geometry with a different-but-valid atom ordering is wrongly rejected.

PROBLEM 67 — LEG_DYSON_EQUATION_BULK *Physics, Condensed Matter Physics* (3 defects)

Original source: (Jain & Allen, 1985)

- **67.3, 67.4, 67.5, 67.6 (step_description_prompt and step_background)** *surface: Broken cross-reference* · neutral-cleanup
  Five prompt and background fields contain dangling cross-references of the form 'as described in step' or 'introduced in step' with the target step number missing, leaving the reader unable to trace the cited dependency.
- **67.5 (step_description_prompt)** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The step description labels the physical constant as hbar/m_e, but hbar/m_e carries dimensions of velocity times length (J.s / kg = m^2/s), which is dimensionally inconsistent with the stated units meV nm^2 (energy times area). The quoted numerical value 76.2 meV nm^2 is consistent with hbar^2/m_e, not hbar/m_e, and the test cases confirm this by using 7.62 meV nm^2 / m_eff (the effective-mass scaling of hbar^2/m_e) for Fermi energy and velocity calculations. Any solution that reads the spec literally and defines the constant as hbar/m_e operates with a quantity of the wrong physical dimension, producing incorrect Fermi energies that cannot match the expected outputs.
- **67.6 (test_cases and general_tests)** *scientific: Over-tight tolerance* · too-strict/wrong
  The test compares an N=101 finite-matrix RPA numerical result against a closed-form analytic expression using a relative tolerance of 1e-15, which is effectively bit-exact and physically unachievable: a finite-difference construction of this size accumulates O(N * eps_machine) rounding error, placing legitimate results near 1e-13 relative error or larger. Simultaneously, the np.allclose checks rely on default tolerances (rtol=1e-5, atol=1e-8) without explicit specification, making acceptance criteria brittle across environments. A correct solution matching the analytic result to any physically meaningful precision is rejected. The fix loosens the numeric-vs-analytic relative tolerance to 1e-8 and adds explicit atol=rtol=1e-10 to the allclose calls.

PROBLEM 68 — HELIUM_ATOM_DMC *Chemistry, Quantum Chemistry* (5 defects)

Original source: (Reynolds et al., 1982)

- **68.3** *scientific: Over-tight tolerance* · too-strict/wrong
  The Laplacian tests assert np.allclose(RMSE, precomputed_target), pinning the finite-difference RMSE to one exact numerical value; any correct implementation with a slightly different rounding path fails. The fix replaces this with the physically meaningful contract that the numerical-vs-analytic RMSE is small (< 1e-2).
- **68.5** *scientific: RNG-dependent grading* · too-strict/wrong
  The Metropolis test asserted exact agreement with a seeded final walker configuration, requiring bitwise reproduction of a specific RNG trajectory; since Metropolis equilibration to |psi|^2 is path-irrelevant, the only physically correct contract is convergence of the mean radius to the known stationary value 3/(2 alpha). The fix replaces the exact-position test with that statistical equilibration check and refactors the function to a single sweep so callers control the loop.
- **68.6** *scientific: Unspecified convention* · too-strict/wrong
  The docstring left drift_new's normalization ambiguous (drift_old explicitly stated 'multiplied by dtau' but drift_new did not) and gave no specification for the return value, so a solver could legitimately return the

clipped min(1, ratio) acceptance probability and fail the gold test. The fix pins the convention that drift_new is also multiplied by dtau and requires the raw uncapped Metropolis-Hastings ratio.

- **68.7** *scientific: RNG-dependent grading* · too-strict/wrong
  The original branching test asserts exact walker indices from a seeded np.random.multinomial draw, which implicitly mandates one specific RNG protocol; any correct alternative such as systematic or residual resampling produces different indices and fails despite satisfying the same physical contract. The true requirement for Diffusion Monte Carlo branching is only that the expected multiplicity of walker i is proportional to w_i / mean(w); the seed-locked index check is a bitwise-reproduction artifact, not a physical correctness criterion. The fix replaces the exact-index assertion with a statistical frequency test over M=20000 draws using deliberately spread weights, so any unbiased resampling scheme passes while degenerate schemes (uniform sampling, w^2 weighting) correctly fail.
- **68.8** *scientific: RNG-dependent grading* · too-strict/wrong
  The four original tests seed the RNG and assert np.allclose on the full energy sequence returned by run_dmc, pinning a specific RNG trajectory that depends on branch-selection order and warm-up details—implementation choices that are physically irrelevant to diffusion Monte Carlo. Any correct DMC implementation that uses a different branching scheme, walker ordering, or warm-up strategy will produce a different stochastic trajectory and fail the exact-sequence check even though its mixed estimator converges correctly to the helium ground-state energy (-2.903724 Ha). The fix replaces the trajectory-pinning assertions with a statistical test: the warm-up-dropped mean local energy must fall within 0.05 Ha of -2.903724 Ha, validated robust across seeds (|dev| ~ 0.008—0.010, dev + 3*SEM <= 0.022), and the prompt and return specification are rewritten to state the mixed-estimator contract instead of prescribing RNG-protocol details.

PROBLEM 69 — LEG_DYSON_EQUATION_SEMI_INFINITE *Physics, Condensed Matter Physics* (2 defects)

Original source: (Jain & Allen, 1985)

- **69.1, 69.2, 69.3 (step_description_prompt)** *surface: Typo / formatting* · neutral-cleanup
  A stray editorial cross-reference annotation (‘[duplicate LEG_Dyson equation-bulk step]’) from the problem-authoring process was left in the solver-facing prompt of steps 69.1, 69.2, and 69.3, polluting the spec with internal bookkeeping text irrelevant to the solver.
- **69.7 (step_description_prompt)** *scientific: Unspecified convention* · too-strict/wrong
  The Raman intensity formula in step 69.7 uses symbols alpha, V, delta, and kd without defining them or mapping them to function inputs (delta_E, kd), so a correct implementation has no basis for choosing the right quantities and can be marked wrong due to the spec gap.

PROBLEM 71 — GADC_REV_COHERENT_INFO *Physics, Quantum Information/Computing* (11 defects)

Original source: (Khatri et al., 2020)

- **71.1** *surface: Interface / return-contract mismatch* · too-strict/wrong
  The function signature ‘def ket(dim)’ omits the ‘args’ parameter that the docstring describes as the basis-vector index, making the header self-contradictory and the function uncallable with the required argument; the output shape (D, 1) was also left unspecified.
- **71.9** *surface: Broken cross-reference* · too-strict/wrong
  The step prompt contains three coupled errors: a broken cross-reference to a nonexistent ‘neg_coh_info in .’ with an empty placeholder; the label ‘coherent information’ instead of ‘channel reverse coherent information’; and the maximizing state amplitude sqrt(p)|00> + sqrt(1-p)|11>, which is swapped relative to the state defined in step 71.8, making the optimization target inconsistent.
- **71.9** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The function GADC_rev_coh_inf computes the channel reverse coherent information I_R = S(B) - S(AB), which swaps the roles of output and environment relative to the standard coherent information I_C = S(B) - S(E). The original docstring and return-variable name labeled the output as “coherent information” / channel_coh_info, misidentifying the quantity as I_C—a distinct quantity with a different formula and generally different numerical value. A correct implementation of I_R that returns the right number would be graded against a label claiming it should equal I_C, causing mismatch and incorrect rejection.

- **71.2** *surface: Interface / return-contract mismatch* · too-strict/wrong
  The signature 'def tensor()' accepts no arguments despite the docstring describing a variadic sequence of matrices; the function cannot receive any input. Output dimensionality behavior (1-D vs 2-D depending on input) was also unspecified.
- **71.3** *surface: Interface / return-contract mismatch* · too-strict/wrong
  The prompt used the same symbol 'i' for both the subsystem index (which subsystem the channel acts on) and the Kraus-operator index K_i, an ambiguous collision that makes the formula uninterpretable without guessing which 'i' is meant where.
- **71.3** *scientific: Unspecified convention* · too-strict/wrong
  The permutation parameter 'perm' was described only as 'desired order', leaving the index base and mapping direction unstated; perm values like 2 and 3 are out of range for 0-based indexing, so only the 1-based convention with output-subsystem-k = input-subsystem-perm[k] reproduces the targets.
- **71.4** *scientific: Unspecified convention* · too-strict/wrong
  The 'sys' parameter for apply_channel did not state whether subsystem indices are 0-based or 1-based; the convention is load-bearing because a test uses sys=[2] on a 2-subsystem state, which is valid only under 1-based indexing.
- **71.5** *scientific: Unspecified convention* · too-strict/wrong
  The 'sys' parameter for partial_trace gave no indication of index base; the 1-based convention governs which subsystems are traced out, so leaving the base unstated renders the interface under-specified and grades implementations using 0-based indexing as wrong.
- **71.6** *scientific: Unspecified convention* · too-strict/wrong
  The prompt gave no formula for von Neumann entropy and left the logarithm base unspecified; entropy in nats versus bits differs by a factor of ln 2, so the missing 'S(rho) = -tr(rho log_2 rho)' formula and explicit 'bits (log base 2)' pin are necessary to grade numerical outputs correctly.
- **71.8** *scientific: Unspecified convention* · too-strict/wrong
  The reverse coherent information formula I_R(A|B) = S(A) - S(AB) references systems A and B without defining which physical qubit each label denotes; the assignment (A = reference qubit, B = channel output) is required to compute the correct marginals S(A) and S(AB).
- **71.8** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The function neg_rev_coh_info computes the reverse coherent information I_R, but the original docstring and output variable were labeled 'coherent information' / 'neg_I_c', the distinct quantity I_C; this mislabeling contradicts the function name and causes models implementing I_R correctly to appear to return the wrong quantity.

PROBLEM 72 — ISING_MODEL *Physics, Condensed Matter Physics* (10 defects)

Original source: (Metropolis et al., 1953)

- **72.1** *surface: Typo / formatting* · neutral-cleanup
  The documented neighbor order (left, above, right, below) is reordered to (above, right, below, left); since all four neighbors are summed, the change is cosmetic and leaves no gold or test behavior altered.
- **72.8** *scientific: RNG-dependent grading* · too-strict/wrong
  The original tests require exact per-temperature mag2 array reproduction under a fixed seed, which fails for any correct implementation that differs in RNG draw order; the replacement validates physics behavior (high magnetization at low T, low magnetization at high T, monotone gap exceeding 0.5) using sufficient equilibration sweeps.
- **72.9** *scientific: Unspecified convention* · too-strict/wrong
  The original spec says to return 'the temperature at which the derivative is minimized,' but the discrete derivative np.diff(mag2)/np.diff(T) is defined on intervals rather than single temperatures, so the instruction is ambiguous about whether to return the left or right endpoint; the correction pins the convention as T_list[argmin(...)] (the left endpoint of the steepest-drop interval).
- **72.9** *scientific: RNG-dependent grading* · too-strict/wrong
  The original tests call scan_T with np.random.seed(0) and a small lattice (N=5, nsweeps=100), then assert np.allclose(calc_transition(Ts, mag2), target): the precomputed target encodes a specific MC RNG trajectory, not the mathematical correctness of the derivative-argmin estimator. Any correct implementation that uses a different RNG state, sweep count, or Metropolis ordering produces a different mag2 array and fails the allclose comparison even though it correctly identifies the critical temperature. The fix replaces these RNG-coupled

tests with a deterministic unit test on synthetic input ([1.0, 1.0, 0.2, 0.1] at T=[1, 2, 3, 4], where the argmin of the discrete derivative is unambiguously T=2.0) and a physics anchor asserting the recovered Tc lies within 0.2 of the exact 2D Ising value 2/ln(1+sqrt(2)) = 2.26919.

- **72.2** *surface: Truncation / garble / dropped constant* · too-strict/wrong
  The prompt is corrupted in two places: the spin value is truncated to '1 or -' (dropping the defining '-1') and the cross-reference to step 1 is garbled to 'given in .', leaving both the spin alphabet and the available helper function unspecified.
- **72.3** *surface: Truncation / garble / dropped constant* · too-strict/wrong
  The prompts for energy-over-all-sites and total-magnetization both end mid-token with 'either 1 or -', omitting the '1' that completes the spin alphabet {+1,-1} and leaving the lattice domain undefined.
- **72.5** *scientific: Unspecified convention* · too-strict/wrong
  The docstring labels the return 'acceptance ratio' without specifying whether the Boltzmann factor is capped at 1; the correct Metropolis probability is min(1, exp(-beta*dH)), and the precomputed test target (54.5982) reflects an uncapped value rather than the correct capped value of 1.0.
- **72.6** *surface: Typo / formatting* · downstream change (not counted as a defect)
  The original prompt wording is minimal and clean; the change only naturalizes phrasing and restates the return value to align with the sweep-chain API redesign, not a defect in the original specification.
- **72.6** *scientific: RNG-dependent grading* · too-strict/wrong
  The original tests pin the exact post-sweep lattice state for a fixed seed, so they pass only if the model reproduces SciCode's specific per-site RNG draw order and short-circuit policy; any correct Metropolis sweep that uses a different but valid traversal order fails, making the tests RNG-protocol over-specifications rather than physics checks.
- **72.7** *surface: Interface / return-contract mismatch* · too-strict/wrong
  The original docstring contradicts itself: the body says the function collects 'iteration, temperature, energy, and magnetization^2 in a dataframe' while the Return field declares 'mag2: (numpy array)', and neither version mentions thermalization; the spec<->return-type inconsistency makes the expected output ambiguous.
- **72.7** *scientific: RNG-dependent grading* · too-strict/wrong
  The original tests require np.allclose on the full per-sweep mag2 trajectory for a fixed seed, passing only when the model exactly reproduces SciCode's RNG draw sequence across all sweeps; the replacement checks instead that the equilibrium mean mag2 lies within 3 blocking-error standard deviations of the expected value, accepting any physically correct Metropolis implementation.
- **72.8** *surface: Interface / return-contract mismatch* · downstream change (not counted as a defect)
  The scan_T signature is updated from (Ts, N, nsweeps) to (Ts, N, n_equil, n_measure) to propagate the burn-in/measure split introduced in 72.7; the original scan_T spec is sound and this is a mechanical API propagation, not an original defect.

PROBLEM 73 — XRAY_CONVERSION_II *Physics, Condensed Matter Physics* (4 defects)

Original source: (Busing & Levy, 1967)

- **73.3 / 73.5 / 73.8 / 73.9 (function_header z_s description)** *scientific: Wrong gold (method / sign / symmetry)* · neutral-cleanup
  The z_s parameter (frame-stepping scan increment) was mislabeled as the phi rotation in four function headers; phi is a fixed diffractometer angle passed separately, so the scan axis is theta. The wrong label misleads implementations into confusing the two rotations.
- **73.2 (function_header det_d description)** *scientific: Unspecified convention* · neutral-cleanup
  The description 'sample distance to the detector' is ambiguous between the perpendicular gap and the along-beam projection; the Q-calculation geometry requires the sample-to-beam-center distance measured along the incident beam (+x direction), which the corrected description specifies.
- **73.6 (step_description_prompt)** *surface: Truncation / garble / dropped constant* · neutral-cleanup
  The prompt defined only the scalar formula d* = 1/d, omitting the enumeration and grouping task: collect all (h,k,l) with d* at or below d*_max and return a dictionary mapping each distinct d* value to its list of reflections, which is what the function signature and tests actually require.
- **73.6 / 73.7 / 73.8 / 73.9 + general_tests (test_cases)** *scientific: Non-discriminating test (physics)* · too-strict/wrong
  For a symmetric cell, powder-ring reflections, indexed hkl, and the U orientation matrix are defined only

up to point-group symmetry (sign flips, permutations, signed-permutation gauge on U). The original exact comparators reject all symmetry-equivalent correct answers; the corrected tests use set-equality on reflections, sign/permutation-invariant |hkl| comparison, and require target.T @ result to be a signed permutation matrix.

PROBLEM 74 — HOUSEHOLDER_QR *Mathematics, Numerical Linear Algebra* (1 defect)

Original source: (Householder, 1958)

- **problem_background_main + sub_steps[0] (step 74.1) step_background** *scientific: Unspecified convention* · too-strict/wrong
  The spec stops after the v-direction formula and never states that all n reflectors k=1..n must be applied; in the square case m=n the final reflector acts on a 1x1 subcolumn as F=[-1], flipping the sign of R's last diagonal entry. A solver that stops at n-1 reflectors (a natural reading of the terse spec) produces R[2,2]=+1.498 instead of -1.498, and is wrongly rejected.

PROBLEM 75 — GRAPHENE_TIGHT_BINDING *Material Science, Semiconductor Materials* (4 defects)

Original source: (Moon & Koshino, 2012)

- **75.1** *surface: Truncation / garble / dropped constant* · too-strict/wrong
  The spec gives the decay constant b as a bare placeholder '(b,a.u.)^{-1}' with no numeric value; without b = 1.17 Bohr^{-1} the exponential hopping V_pp ~ exp(-b * distance) cannot be evaluated, making correct solutions unreproducible. The unit string 'b, a.u.' was also garbled, conflating the variable name with the unit.
- **75.2** *surface: Broken cross-reference* · neutral-cleanup
  The phrase 'using the hopping evaluation from .' contains an empty reference target, leaving the required function unidentified; the correction names step 1 (hopping_mk) as the source, resolving the dangling cross-reference.
- **75.2** *scientific: Unspecified convention* · too-strict/wrong
  The parameter description for di, dj does not state they are integer unit-cell indices (n1, n2) or how the displacement vector is formed; the convention d = basis[ai] - (basis[aj] + n1*latvecs[0] + n2*latvecs[1]) is load-bearing, and applying the shift to the wrong atom yields incorrect bond lengths that fail the gold test.
- **75.3** *scientific: Unspecified convention* · too-strict/wrong
  The prompt omits both the hop-inclusion cutoff (in-plane projected distance <= a0 = a/sqrt(3)) and the Bloch-sum formula H_ij(k) = sum_R (-t(d)) e^{ik.d}; without the nearest-neighbor cutoff the Hamiltonian includes spurious long-range hops, shifting eigenvalues from the gold values [-3.258, -2.117, 2.472, 2.903] and rejecting any correctly implemented general solver.

PROBLEM 76 — PROTEIN_DNA_BINDING *Biology, Genetics* (5 defects)

Original source: (Schneider et al., 1986)

- **76.1 (problem_description_main; sub_steps[0].step_description_prompt; sub_steps[0].function_header)** *scientific: Unspecified convention* · too-strict/wrong
  The original spec declares each PWM row a probability distribution (implying L1 normalization) yet types the output as an integer array, which is self-contradictory; the correction pins L2 (Euclidean) per-row normalization with a float dtype, giving a single coherent convention.
- **76.2 (sub_steps[1].step_background)** *scientific: Unspecified convention* · too-strict/wrong
  The KL-divergence formula uses the ambiguous symbol log, but the gold value (7.4987...) is reproduced only with the natural logarithm; using log base 2 or log base 10 yields a different result, so the unspecified base rejects correct solvers choosing a non-natural base.
- **76.4 (sub_steps[3].function_header)** *scientific: Spec↔gold contradiction* · too-strict/wrong
  The docstring promises an integer return value, but the tests assert None when no binding site is found; a correct implementation following the docstring returns an integer sentinel and fails those tests solely due to the spec-test contradiction.

- **76.4 (sub_steps[3].test_cases; general_tests)** *scientific: RNG-dependent grading* · too-strict/wrong
  The scanner aggregates num_runs=100 random samplings and returns the most-frequent position, so demanding exact equality to a single hard-coded target ties correctness to a specific RNG draw; the corrected test checks that the returned position falls within one motif-length of the known inserted position, rejecting wrong answers without over-specifying the stochastic procedure.
- **76.4 (sub_steps[3].test_cases; general_tests)** *surface: Trivially broken test* · too-strict/wrong
  Tests for data2 and data3 generate DNA sequences with binding sites from those motifs but scan using load_motif_from_df(data), a copy-paste mismatch; a correct scanner cannot detect a data2-inserted site by scanning with the data motif, so correct implementations fail these tests due to the wrong motif being loaded.

PROBLEM 77 — BERENDSEN_THERMOSTAT *Material Science, Molecular Modeling* (10 defects)

Original source: (Berendsen et al., 1984)

- **77.1 (problem_io) and 77.12 (velocityVerlet function_header)** *surface: Truncation / garble / dropped constant* · neutral-cleanup
  The P_target unit line is garbled by a copy-paste artifact—'units: bar.ostat. Set to 0 to deactivate, units: picoseconds.'—fusing parts of two different parameter descriptions; the correction reads simply 'units: bar.'.
- **77.12 (velocityVerlet step_description_prompt)** *scientific: Unspecified convention* · too-strict/wrong
  The Berendsen barostat box-scaling factor requires an isothermal compressibility value, but the original spec omits it entirely; without the specific constant (gamma = 4.6e-5 bar^-1, water's value), any general correct implementation produces an unmatchable gold.
- **77.3 (dist_v function_header)** *scientific: Unspecified convention* · too-strict/wrong
  The dist_v docstring declares a scalar float return ('minimum image distance') but the function actually returns the minimum-image displacement vector from r1 to r2; the wrong return type and missing direction convention cause downstream sign errors in force calculations.
- **77.5 (f_ij function_header)** *surface: Interface / return-contract mismatch* · too-strict/wrong
  The f_ij docstring types argument r as a scalar float distance, but the function receives a 3D displacement vector; the 'from particle i to particle j' direction is load-bearing for the sign of the computed force.
- **77.6 (E_tail step_background)** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The Lennard-Jones energy-tail formula in the step background omits the required 1/V (V = L^3) normalization factor: the correct total tail correction is (8/3) pi N^2 eps sigma^3 [...] / V, but the original writes it without the /V and mislabels it "per particle." Because the N^2 prefactor already counts all pair interactions, dividing by V converts the extensive sum to an intensive bulk correction; omitting it inflates the result by a factor of V (a factor of 1000 for a typical 10^3 simulation box). Any correct implementation that includes the 1/V factor therefore produces a value differing from the precomputed gold by three orders of magnitude and is incorrectly rejected.
- **77.7 (P_tail step_background)** *scientific: Wrong gold (method / sign / symmetry)* · too-strict/wrong
  The long-range pressure tail correction for a Lennard-Jones fluid is P_tail = (16/3) * pi * (N^2/V^2) * epsilon * sigma^3 * [(2/3)(sigma/r_c)^9 - (sigma/r_c)^3], where the N^2/V^2 factor reflects the pair density squared integrated over the tail volume. The original formula omitted the 1/V^2 (V = L^3) factor entirely, making the precomputed gold too large by a factor of V^2. Any correct implementation that includes 1/V^2 therefore produces a pressure differing from the gold by orders of magnitude for typical simulation box sizes, causing the step to fail regardless of the solution's physical correctness.
- **77.7 (P_tail function_header)** *scientific: Unspecified convention* · too-strict/wrong
  P_tail must return pressure in bar, but the unit conversion from zeptojoules per nm^3 to bar depends on knowing that L, sigma, and rc are in nanometers and epsilon is in zeptojoules; omitting these units from the docstring leaves implementations unable to derive the x10 conversion factor.
- **77.9 (temperature test_cases)** *surface: Trivially broken test* · neutral-cleanup
  The N=1 temperature test passes a 1D shape-(3,) velocity array where the function's contract is shape-(N,3); the fix reshapes it to (1,3) for contract consistency, though the numerical result is unchanged since np.sum(v**2) is shape-invariant here.

- **77.10 (pressure step_background)** *scientific: Unspecified convention* · too-strict/wrong
  The virial-pressure formula defines r_ij as 'displacement from particle i to j' (= r_j - r_i), but the correct virial convention requires r_ij = r_i - r_j; using the wrong sign flips the sign of the virial pressure, turning a correct +8.851 result into -8.851.
- **77.10 (pressure test_cases) and 77.11 (forces test_cases)** *surface: Trivially broken test* · too-strict/wrong
  In the second test case of both the pressure (77.10) and forces (77.11) steps, N is declared as 2 but the positions array has three rows, making the test ill-defined; the fix removes the spurious third row so the array shape matches N.

PROBLEM 79 — NOSE_HOOVER_CHAIN_THERMOSTAT *Material Science, Molecular Modeling* (8 defects)

Original source: (Martyna et al., 1996)

- **problem_io / sub_steps[3] (step 79.4) function_header** *surface: Interface / return-contract mismatch* · too-strict/wrong
  The spec declared position and velocity trajectories as shape (nsteps, 1), but the gold comparison expects 1-D arrays of shape (nsteps,); a solver following the stated shape literally produces a 2-D output and is wrongly rejected.
- **sub_steps[3] (step 79.4) function_header** *scientific: Unspecified convention* · too-strict/wrong
  The original header gave no convention for trajectory indexing, leaving ambiguous whether the first entry is the initial state or the state after one step; the correction pins the record-then-step convention (x[0]=x0, v[0]=v0), which is load-bearing for matching the gold trajectory.
- **sub_steps[1] (step 79.2) and sub_steps[2] (step 79.3) function_header** *surface: Interface / return-contract mismatch* · too-strict/wrong
  The Nose-Hoover chain of length M has per-link forces G_i, thermostat velocities v_xi_i, and positions xi_i, each a vector of shape (M,); declaring them as scalar float contradicts the multi-link algorithm and causes a correct array-based implementation to be rejected.
- **sub_steps[1] (step 79.2) and sub_steps[2] (step 79.3) function_header** *scientific: Unspecified convention* · too-strict/wrong
  The original docstring described dt only as 'the integration time step', concealing that nhc_step advances the chain by half its argument; solvers following the Liouville framing (exp(iL w_j dt/2)) passed w_j*dt/2 instead of w_j*dt, producing a factor-of-2 error in integration time.
- **sub_steps[1] (step 79.2) step_description_prompt** *scientific: Unspecified convention* · too-strict/wrong
  The prompt used k_B and Q_k in the G_1/G_k formulas without fixing their numerical values; multiple choices (e.g. Q=1 or k_B=2) each fail the targets, while only k_B=1 and Q_k=k_B*T/omega^2 reproduce the gold, making the step ill-posed without these conventions.
- **sub_steps[2] (step 79.3) step_description_prompt** *scientific: Unspecified convention* · too-strict/wrong
  Naming 'Yoshida's fourth-order method' is insufficient because multiple 4th-order composition schemes exist with different coefficients; without pinning the triple-jump weights w1=1/(2-2^(1/3)), w2=1-2*w1, the gold trajectory is ambiguous and correct implementations using a different valid scheme are wrongly rejected.
- **sub_steps[0] (step 79.1) test_cases** *scientific: Non-discriminating test (physics)* · too-lenient (test tightened)
  All three original Verlet test cases set x0=0.0, so an implementation that mishandles the initial-position term in the position update passes regardless; adding a case with x0=1.0 exercises the position dependence and closes this false-pass hole.
- **sub_steps[3] (step 79.4) test_cases / general_tests** *scientific: Over-tight tolerance* · too-strict/wrong
  The Nose-Hoover chain (NHC) integrator is chaotic: for M=2 chains, trajectories computed with N=20000 and N=40000 steps accumulate exponentially diverging floating-point rounding errors across any two correct-but-not-bit-identical implementations. The original test compared full trajectories element-wise against a precomputed gold, so a correct solver whose intermediate floats differ by machine-epsilon amounts is rejected at step counts far beyond the Lyapunov time. Reducing the M=2 cases to N=500 and N=1000 keeps the

integration within the regime where correct implementations remain mutually reproducible under exact comparison, eliminating false failures without weakening coverage of the two-thermostat chain.

PROBLEM 80 — ANDERSON_THERMOSTAT *Material Science, Molecular Modeling* (5 defects)

Original source: (Andersen, 1980)

- **problem_io / step 80.7 (MD_NVT function_header)** *scientific: Spec↔gold contradiction* · too-strict/wrong
  The spec described a Berendsen thermostat plus barostat (NPT ensemble) and claimed a modified box length is returned, directly contradicting the problem's actual algorithm (Andersen NVT, no barostat) and the true five-element return; a solver following the original spec would implement the wrong ensemble and wrong return contract.
- **step 80.2 (E_ij) and step 80.4 (f_ij) function headers** *surface: Interface / return-contract mismatch* · neutral-cleanup
  The E_ij and f_ij docstrings mentioned a truncated-and-shifted Yukawa potential as an optional component, but no Yukawa parameters exist anywhere in the task; the spurious mention misleads solvers without affecting grading, since the gold tests only exercise Lennard-Jones.
- **step 80.3 (background), step 80.4 (f_ij r-type), step 80.5 (forces Returns)** *scientific: Unspecified convention* · too-strict/wrong
  The background limited energy to 'two atoms next to each other' instead of all distinct pairs (i<j) at minimum-image distance; f_ij's argument r was typed as a scalar float when the gold requires the signed 3D displacement vector r_j - r_i; and the Newton's-third-law force accumulation sign convention was left implicit, so correct vectorial implementations could be graded wrong.
- **step 80.5 (forces) test_cases** *surface: Trivially broken test* · too-strict/wrong
  The forces() test declared N=2 but supplied a three-row positions array; a correct implementation iterating over N=2 would silently ignore the third particle, making the test input self-contradictory and the expected output physically wrong.
- **step 80.7 (MD_NVT) test_cases and general_tests** *scientific: RNG-dependent grading* · too-strict/wrong
  N=200 conflicts with initialize_fcc, which returns 6^3=216 lattice sites, so the declared particle count mismatches the actual positions array; additionally, the Andersen thermostat makes stochastic collision draws, so without a fixed RNG seed the temperature and energy assertions are non-reproducible across runs.