Backward and contrary checks caught 39 of 47 quiet faults in eight small simulated instruments

A preregistered mutation study of reference-free tests against a wide forward test, with the seven faults that passed everything

Zain Dana Harper

Version 1.0, 14 November 2026. Finding F204 in the findings ledger. Signal (tier B). Simulation only.

Abstract

A numerical instrument model is usually tested forward: feed it a case with a known answer and compare. We asked how much two checks that need no known answer add: running the model backward against itself, and running it against its contrary. The study is a simulation. Code planted one fault at a time at every arithmetic site of eight small instrument models we wrote, and scored which tests noticed. Among 47 single faults that passed one calibration point, the backward and contrary checks together caught 39 (83 percent) and a wide forward test caught 29 (62 percent); sign-flip p 0.003, corrected to 0.013 within the study and 0.009 across the 84 tests of the nine-study set. The advantage sits in three of the eight instruments, and one of them, the Radon projector, supplies 23 of the 47 faults; without it the counts are 16 and 12 of 24. Seven faults passed every test: four divide by a grid spacing that every test set to 1, and three leave a wrong wave propagator that still inverts itself exactly. On pairs of faults the order reversed with no clear difference: the wide test caught 27 of 30 and the backward and contrary checks 24 of 30, p 0.12. Nothing here was measured on hardware or on code written by anyone else.

1 What this means for a reader

If you maintain code that models a physical instrument, a test that runs the model forward and then backward costs little and needs no trusted answer, and in this simulation it found faults that many forward reference points missed. It has a blind spot that matters: a model that is wrong in a self-consistent way inverts itself perfectly and passes. So keep at least one absolute reference, and never calibrate with every scale factor set to 1. The numbers come from eight small models and planted faults, and a test on a real code base with real bugs is what would confirm or sink them.

2 Introduction

Scientific code often has no cheap source of right answers. Software testing calls this the oracle problem, and metamorphic testing is the standard response: check a relation that must hold between several runs of the program. The term comes from software engineering and covers any test that compares outputs of related inputs with each other; a test that compares one output with a stored expected value is outside it. Segura and colleagues surveyed 119 papers on the method [1], and a later article by the same first author warns that a satisfied relation does not confirm that an output is right [2].

Physics supplies such relations for free. A lossless film transmits the same from either side. A wave run forward and then backward in time returns to its start. A projector and its adjoint satisfy an inner-product identity. The adjoint, in linear algebra and inverse problems, is the partner map that moves a vector from the output space back to the input space so that the two inner products agree; it is a different thing from the inverse, which undoes the map. Claerbout's dot-product test, which compares two scalars computed with random vectors, has been a routine check of such pairs in geophysical imaging for decades, and his texts say a failure condemns

the code while a pass leaves correctness unproven [3]. A recent tutorial on adjoint gradients names the same blind spot we met here: the test checks mutual consistency, so a wrong derivative with a matching wrong transpose passes [4].

None of this is new, and we claim no novelty. The question was a practical one for our own work: how much weight a reference-free check should get beside a forward test when a new numerical kernel is written. The decision it informed was whether the check enters our catalogue of default checks. We priced it with mutants. A mutant, in mutation testing, is a copy of a program with one small fault planted by rule; it is a stand-in for a bug and may not resemble the bugs people write. The survey above describes the share of mutants a test detects as a common measure of test strength [1].

3 Preregistration and design

The study ran twice on 31 October 2026, each time under a preregistration: a plan with forecasts and rules, hash-locked before the run.

The first run compared four test sets on every mutant of every function. It gave the backward and contrary checks 79 of 89 false passes (89 percent) against 29 of 89 (33 percent) for the wide forward test. Reading the 51 faults that only the reference-free checks caught showed that most sat in functions the forward tests never called, or in an output the forward film test never read. The result was partly a gap in our own forward tests. That run is kept as it ran, and its headline is not claimed here.

The second preregistration was written after the first result was seen, and says so in its text. It scores only faults in functions that the forward tests call, makes the forward film tests read both outputs, reports faults in inverse functions apart, and adds pairs of faults. Its forecasts were that the wide forward test would catch more than the reference-free checks, on single faults and on pairs. Its kill condition, the outcome fixed in advance that would end the claim, was a catch rate under 30 percent for the reference-free checks on single faults. Each forecast became a rule that code reads off the result as a verdict.

The first preregistration also covered a re-reading of two earlier studies of ours with the same three-way idea. That part was post hoc and is not reported in this paper.

4 Data and provenance

There is no external dataset. The material is eight instrument models written for this study in plain Python with NumPy, each a few lines long so that every arithmetic site can be changed: a normal-incidence thin film by characteristic matrices; an angular-spectrum wave propagator; a leapfrog wave equation in one dimension; a blur with its adjoint and a regularised deblur; a pixel-driven Radon projector with its adjoint; an explicit heat-diffusion step; ray transfer matrices; and a Stokes analyzer row with a least-squares inversion. The faults were generated by rule from that source. The forward reference values are closed-form results from textbook physics, coded by us. All of it is our own work, written the same day as the tests.

5 Methods

Planting faults

A program walked the syntax tree of each model and listed every site of seven kinds: a plus or minus, a times or divide, a unary minus, a complex conjugate, a numeric constant (twice: add 1, and double), and an inequality. Each mutant changes one site: plus and minus swap, times and divide swap, the unary minus or the conjugate is dropped, the constant changes, the inequality flips. We did not choose which faults to plant.

A mutant can leave the program's behaviour unchanged; mutation testing calls that an equivalent mutant, and it is no fault at all. A differential oracle separated the two cases: the mutant and the original ran on random

inputs wider than any test uses, and any difference in output beyond a relative tolerance of 1e-9, or a change of array shape, marks a fault. This is a comparison of two programs and says nothing about whether the original is right.

Four test sets

Each instrument has four tests, written before any mutant was scored and adjusted until the original code passed all of them.

- The calibration point: one forward case with a closed-form answer, such as the reflectance of a quarter-wave film. It stands for the single check an instrument often gets.
- The wide forward test: 4 to 15 forward cases with closed-form answers at random parameter values.
- The backward check: the model against itself in reverse. Transmission through a film stack from the other side; a wave propagated out and back; a wave run on with its last two states swapped; blur then deblur; the inner-product identity for the projector and its adjoint; ray matrices for the elements in reverse order; analyzer rows inverted to recover a known Stokes vector. Diffusion has no time reversal, so one small step with the diffusivity negated stood in for it.
- The contrary check: the model against its opposite or complement. A half-wave absentee layer that must leave reflectance unchanged, with energy conserved; an aperture and its complement summing to the open field (Babinet's principle); source and receiver swapped; a lens followed by its negative; crossed polarizers reading zero.

The last two use no reference value. The three-way test is our name for the backward and contrary checks taken together after a forward calibration: a fault is caught three-way when either one fails.

The score

A false pass is a mutant that the oracle marks as a fault and that still passes the calibration point. These are the quiet faults. For a test set that fails on k of the N false passes, the catch rate is

$$c = \frac{k}{N} \tag{1}$$

Mutants that crashed any test left the count, since a crash is noticed without a test.

The two rates were compared on the same faults. For each false pass let d_i be 1 when only the three-way test caught it, minus 1 when only the wide test caught it, and 0 otherwise. A sign-flip test drew 4,000 random sign patterns for the d_i and counted how many gave a mean at or above the real one:

$$p = \frac{j + 1}{n + 1} \tag{2}$$

with j that count and n = 4,000. The four p values of the run (each direction, on singles and on pairs) were corrected with the Benjamini-Hochberg correction [5], which limits the expected share of false discoveries among the tests called significant, and again across all 84 tests of the nine simulation studies run together that day.

Pairs

Two errors can cancel at a calibration point. The run drew 400 random pairs of sites, each pair inside one forward function, from a fixed seed, planted both faults and scored the result the same way. Draws from a function with a single site were skipped, which left 344 pairs.

False alarms

A reference-free check that fails on a correct program would be useless. Two things bound that: every test passes on the original code, and in the first run neither test failed on any equivalent mutant. With 0 failures in

m = 9 equivalents, the one-sided 95 percent upper bound on the false-alarm rate is

$$u = 1 - 0.05^{1/m} \tag{3}$$

which is 0.28.

6 Controls and planted checks

The control permuted the fault labels among the mutants of each instrument 200 times and recomputed the three-way catch rate. It gave a mean of 0.726 with a 95th percentile of 0.76, below the real 0.830 (p 0.005). The preregistration calls this a weak check, and it is: only 9 of the 148 scored mutants are equivalent, so permuting labels changes little.

The first run carried a planted check: with the conjugate dropped in the blur adjoint, the forward tests had to pass and the contrary check had to fail. Both happened.

All four tests passed on the original code of all eight instruments in both runs.

7 Results

The second run planted 200 single faults in the eight forward functions. Of these, 52 crashed a test and left the count, 148 were scored, 9 were equivalent, and 47 were faults that passed the calibration point.

Among 47 single faults that passed one calibration point, the backward and contrary checks together caught 39 (83 percent) and a wide forward test caught 29 (62 percent); sign-flip p 0.003, corrected to 0.013 within the study and 0.009 across the 84 tests of the nine-study set. Taken alone, the backward check caught 33 and the contrary check 28. Eleven faults fell only to the three-way test and 1 only to the wide test. All three test sets together caught 40. Figure 1 gives the rates.

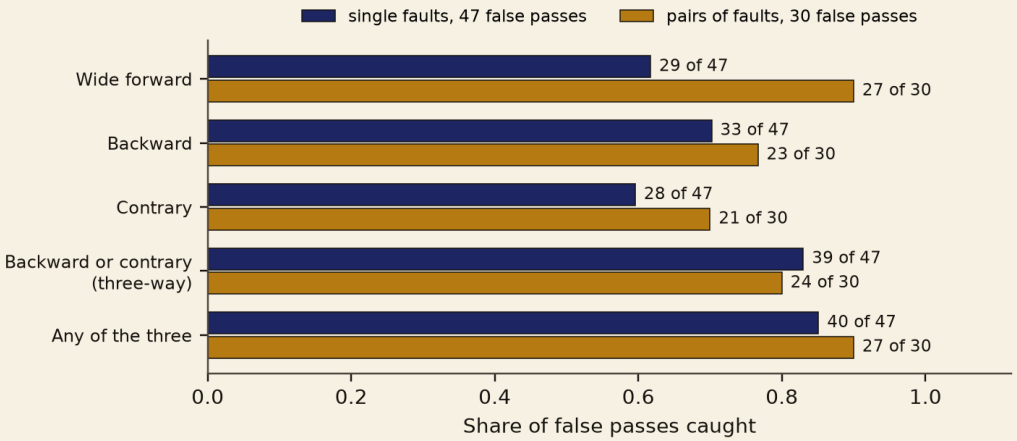


Figure 1. Share of false passes caught by each test set, for single faults and for pairs of faults. Drawn from the second run's result file.

The total hides how uneven the instruments are. The advantage sits in three of the eight instruments, and one of them, the Radon projector, supplies 23 of the 47 faults; without it the counts are 16 and 12 of 24. The split without the projector was computed after the result was seen. Table 1 and Figure 2 give every instrument. In three instruments the three-way test caught nothing.

Seven faults passed every test: four divide by a grid spacing that every test set to 1, and three leave a wrong wave propagator that still inverts itself exactly. The four are the single false pass in the wave code and all

Table 1. Single faults that passed the calibration point, by instrument, and how many each test caught.

Instrument	False passes	Wide forward	Three-way
Thin film	6	4	6
Angular spectrum	3	0	0
Wave, leapfrog	1	0	0
Blur	0	0	0
Radon projector	23	17	23
Heat diffusion	3	0	0
Ray matrices	4	4	4
Stokes analyzer	7	4	6
All eight	47	29	39

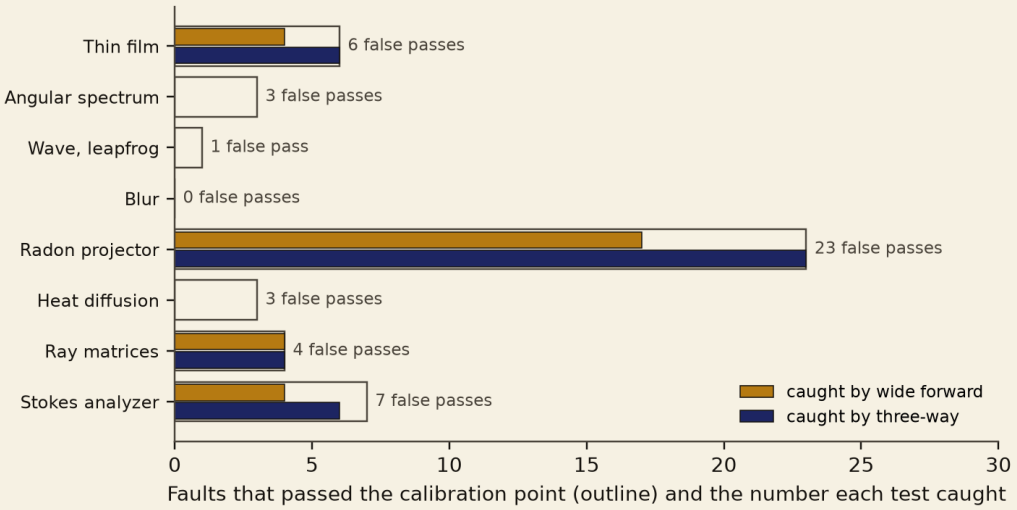


Figure 2. Single faults by instrument. The outline is the number of false passes; the filled bars are the numbers caught. Drawn from the second run's result file.

three in the heat code; each swaps a division by the grid spacing for a multiplication, which changes nothing when the spacing is 1. The three in the angular-spectrum code are two sign changes inside the term under the square root and one division by the transfer function where the code should multiply. Each of those leaves a propagator that conserves power, obeys Babinet's principle and undoes itself when run backward, so the reference-free checks pass, and the forward tests read only a beam width that the fault does not change. The one-step undo that stood in for time reversal in the heat code caught none of its 3 false passes.

On pairs of faults the order reversed with no clear difference: the wide test caught 27 of 30 and the backward and contrary checks 24 of 30, p 0.12. Of the 344 pairs, 155 crashed, and 30 of the remaining 189 were faults that passed the calibration point. Three fell only to the wide test, none only to the three-way test, and 3 passed everything.

Faults in the inverse and adjoint functions were scored apart. All 34 scored mutants there were false passes, the three-way test caught 32 of them and the wide forward test caught 0. The zero could not have come out otherwise, because no forward test calls those functions. The 32 show only that a pairing test reaches code a forward test does not.

Table 2 sets the forecasts against the outcomes. The forecast that the wide test would win on single faults was wrong.

Table 2. Forecasts in the second preregistration against what the run gave.

Quantity	Forecast	Outcome
Single faults passing the calibration point	25	47
Wide forward catch rate, singles	60 percent	62 percent
Three-way catch rate, singles	55 percent	83 percent
Singles caught only by the wide test	at least 3	1
Singles caught only by the three-way test	at least 3	11
Pairs passing the calibration point	50	30
Wide forward catch rate, pairs	75 percent	90 percent
Three-way catch rate, pairs	65 percent	80 percent

8 Discussion

The two kinds of test fail in different places. A forward test sees only the outputs it reads at the parameter values it visits; the thin-film and projector faults it missed were visible to a symmetry. A symmetry test sees only departures from the symmetry; a model that is wrong and still symmetric is invisible to it. The seven survivors are the useful part of the result, because each names a habit to change. Four of them existed only because every test used a grid spacing of 1, where multiplying and dividing agree. Three existed because a propagator with the wrong dispersion is still unitary.

That gives two working rules, which we adopted: calibrate at values other than 1, and keep one absolute reference beside any reference-free check. The catalogue entry for our own kernels now asks for both kinds of test.

The size of the advantage should not be carried anywhere. It depends on which instruments are in the sample, and one instrument holds half the faults. On pairs of faults, where two errors cancel at the calibration point, the wide test did at least as well.

9 Limitations

- This is a simulation on planted faults. Mutants are arithmetic slips made by rule, and no study here shows that they resemble the faults people make in instrument code.
- The eight instruments are small models we wrote ourselves, and we wrote the tests too, on the same day. A stronger forward test or a weaker symmetry test would move both rates.
- One instrument, the Radon projector, supplies 23 of the 47 faults. The three-way test caught 0 of the 7 faults in the angular-spectrum, wave and heat code.
- 52 of 200 single mutants and 155 of 344 pairs crashed a test and left the count, so the rates describe quiet faults only.
- The false-alarm rate is bounded loosely: 0 failures on 9 equivalent mutants allows a rate up to 0.28.
- The second design was written after the first result was seen. It was made to remove an advantage the first design gave the three-way test, and it is still a design chosen with knowledge of the data.
- The split without the Radon projector is post hoc.
- The label-permutation control is weak, as the preregistration says.
- Scoring was by our own code and was not blind. Nobody outside the project has checked the code or the result.

10 Verified

Verified: on 14 November 2026 the counts of mutants, crashes, equivalents and false passes, every catch rate for single faults, pairs and inverse functions, the per-instrument rows, the four p values with their corrected

values, the control's mean, percentile and p, the seven verdicts and the list of seven surviving sites were re-read from the second run's result file; the first run's 89 false passes, its two rates, the 51 faults and the 0 of 9 false alarms were re-read from the first run's result file; and the corrected value across 84 tests was re-read from the study set's correction file. They match the findings ledger, whose q of 0.009 is the across-studies value. The counts printed as "39 of 47" and the like are the file's rates times its denominators. The seven surviving sites were located in the instrument source by listing its mutation sites with the study's own site finder, and the grid spacing of 1 was read in the test source. The preregistration's SHA-256 matches the hash in the result file, the lock file and the sidecar file. Both figures were drawn by a script that reads only the second run's result file. The study was not re-run for this paper. The literature was found by two searches on 14 November 2026 and read at the level of search snippets; the full texts were not read. The Benjamini and Hochberg paper is cited for the method only.

11 Corrections

- 31 October 2026: the first run's result, 89 percent against 33, was traced to forward tests that did not reach the inverse functions or the film's transmittance. The second run was preregistered to remove that, and its numbers replace the first run's.
- 14 November 2026: the ledger entry gives one corrected value, 0.009; this paper prints both the within-study value, 0.013, and the across-studies value, 0.009, and says which is which.
- 14 November 2026: the per-instrument split was added beside the headline count, because the three-way test caught none of the false passes in three instruments.
- 14 November 2026: the re-reading of two earlier studies, which the ledger entry lists, was left out because it was post hoc and needs those studies to be read.
- 14 November 2026: first published version.

12 Reproduction

- Data: no external dataset. The inputs are the eight instrument models and four test sets per model described in the Methods section, written for the study in Python with NumPy; they are not yet in a public repository.
- Plant every single-site fault of the seven kinds in each forward function. Mark faults with a differential oracle on random inputs. Drop mutants that crash a test.
- Keep the faults that pass the calibration point and compute eq. 1 for the wide forward test, the backward check, the contrary check, and the last two together.
- Compare the wide and three-way tests with eq. 2 on 4,000 sign patterns, and correct the p values by Benjamini-Hochberg.
- Draw 400 pairs of sites within one forward function from seed 20261106 and score them the same way. Score the inverse and adjoint functions apart.
- Preregistration SHA-256: d82bfea0f79dd4708e1fe0c06639da6091c977f6e3d10ab07bc11bb7c2803940. Plan SHA-256: be230e2b62c03e607781e72f210fc6023865d5557bd5680f8f2ce02901409918. Run 20261031T110819Z.
- First run, kept as run: preregistration SHA-256 480be6fe4423efd197e7b83994d0cf41696b39fa474565c972d9a4f54bf745e9, run 20261031T110646Z.
- Cost: the analysis took about 3 s on one desktop CPU.

13 Appendix A. The preregistered verdicts

Each verdict is a rule written before the second run. Values are read from that run's result file. Five rules hold and two fail. The pairs rule holds on the point values alone; the difference behind it has p 0.12.

Table 3. Preregistered verdicts of the second run and the rules that decided each one.

Verdict	Deciding rule	Value	Holds
Kill condition not met	three-way catch rate on singles at least 0.3	0.83	yes
Original code clean	every test passes on the original code	all pass	yes
Wide test has its own catches	singles caught only by the wide test, at least 3	1	no
Three-way test has its own catches	singles caught only three-way, at least 3	11	yes
Wide wins on singles	wide rate above three-way rate	0.62 against 0.83	no
Wide wins on pairs	wide rate above three-way rate	0.90 against 0.80	yes
Beats the label control	three-way rate above the control's 95th percentile	0.83 against 0.76	yes

14 Glossary

Each term is defined where it is first used in the text. The same definitions, with the field each comes from and what each excludes, are collected here.

- **metamorphic testing**. Testing a program by checking a relation that must hold between the outputs of several related runs, used when no trusted expected output is available. Field: software engineering (testing). Excludes a test that compares one output with a stored expected value, and any proof that an output is correct.
- **adjoint**. The partner of a linear map that carries vectors from the output space back to the input space so that the inner product of an output with a vector equals the inner product of the input with the partner's result. Field: linear algebra and inverse problems. Excludes the inverse, which undoes the map; an adjoint exists for maps that cannot be undone.
- **mutant**. A copy of a program with one small fault planted by rule, such as a plus swapped for a minus, used to measure whether a test notices. Field: software engineering (mutation testing). Excludes a bug written by a person; mutants stand in for such bugs and may not resemble them.
- **preregistration**. A dated written plan for a study, fixed and hash-locked before any result is seen: the question, the data, the statistic, the controls, the forecasts and the kill conditions. Field: research method (metascience). Excludes a plan revised after looking at the outcome, and a protocol written only for the record after the run.
- **kill condition**. A rule written in the preregistration that, when met, decides against the claim with no room for judgement after the run. Field: research method (our usage, after Popper's falsification). Excludes reasons found after the run for setting a result aside.
- **verdict**. The pass or fail of one preregistered rule set against the run's numbers, read off by code. Field: research method (our usage). Excludes an interpretation written after reading the numbers.
- **equivalent mutant**. A mutant whose change leaves the program's behaviour the same on every input, so that no test can or should fail on it. Field: software engineering (mutation testing). Excludes a fault that a weak test happens to miss.
- **differential oracle**. A judge of behaviour that runs two versions of a program on the same inputs and reports whether their outputs differ; here it marks a mutant as a fault when it differs from the original on random inputs. Field: software engineering (testing). Excludes any statement that the original program is right.
- **calibration point**. A single forward test case with a known closed-form answer, standing for the one check a new instrument model often gets. Field: our usage, from instrument practice. Excludes a calibration of hardware against a physical standard.
- **wide forward test**. A set of 4 to 15 forward test cases with closed-form answers at random parameter values, all of which must pass. Field: our usage. Excludes any check that runs the model backward or against a

complement, and any reference taken from measurement.

- **backward check**. A test that runs an instrument model in reverse against itself, by time reversal, reciprocity, an adjoint identity or an inverse, and needs no reference value. Field: our usage, from physics symmetries. Excludes a comparison with a known answer.
- **contrary check**. A test that runs an instrument model against its opposite or complement, such as an aperture and its complementary screen or a lens and its negative, and needs no reference value. Field: our usage, from physics symmetries. Excludes a comparison with a known answer, and the backward check, which reverses the same case.
- **three-way test**. The backward and contrary checks taken together after a forward calibration; a fault is caught three-way when either check fails. Field: our usage. Excludes the wide forward test, which is scored apart.
- **false pass**. A planted fault, confirmed by the differential oracle, that still passes the calibration point. Field: our usage. Excludes equivalent mutants, and mutants that crash a test.
- **sign-flip test**. A paired test that gives each difference a random sign many times and counts how often the mean of the re-signed differences reaches the real mean. Field: statistics (permutation tests). Excludes a test that assumes a normal distribution, and any comparison of unpaired groups.
- **Benjamini-Hochberg correction**. An adjustment of a family of p values that limits the expected share of false discoveries among the tests called significant. Field: statistics (multiple testing). Excludes control of the chance of any single false discovery, which is the stricter Bonferroni aim.
- **compensating pair**. Two planted faults in one function, drawn at random, kept in the score when together they still pass the calibration point. Field: our usage. Excludes pairs chosen by hand to cancel.
- **control**. A version of the data with the structure under test removed (for example shuffled or pooled), scored the same way as the real data, so the real score has something to beat. Field: statistics and experimental design. Excludes a comparison with a different dataset or with a published number.
- **planted check**. Data built so that the right answer is known, run through the same analysis before the real data, to show that the analysis can see the effect when it is there. Field: research method; called a positive control in experimental science. Excludes any check made on the real data, which cannot show what the analysis misses.
- **findings ledger**. The append-only, dated list of every result we have recorded, numbered F1, F2 and on; corrections are added as new dated lines and old entries are never rewritten. Field: our records. Excludes a curated list of successes.

15 References

[1] Segura, S., Fraser, G., Sanchez, A. B. and Ruiz-Cortes, A. (2016). A survey on metamorphic testing. IEEE Transactions on Software Engineering.

[2] Segura, S. and colleagues (2020). Metamorphic testing: testing the untestable. IEEE Software.

[3] Claerbout, J. F. (2010). Basic Earth Imaging. Stanford Exploration Project. Section "Dot product test".

[4] From inverse problems to adjoint gradients: a discrete, implementation-oriented tutorial on PDE-constrained optimization. arXiv:2610.04789.

[5] Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society, Series B 57, 289-300.