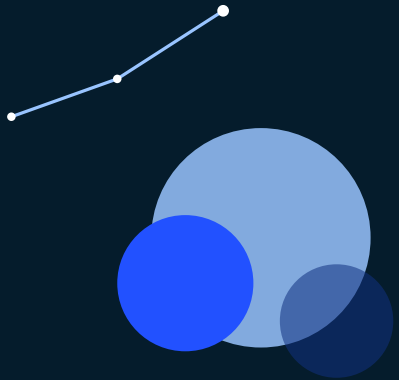
PYHEP.DEV 2026 · WEDNESDAY MORNING · 9 SEPTEMBER 2026

Rebuilding a legacy Dalitz-plot fitter

what machine-checked parity caught

Josué Molina

Independent Researcher · Veltman Centre, Nikhef



A quiz. Which of these is wrong?

(a) A published fit-fraction budget

$$\sum_i \mathrm{FF}_i = 112.8\,\%$$

$$\sum_{i<j} \mathrm{FF}_{ij}\ \text{(printed)} = -6.33$$

(b) A spline-segment lookup, 41-knot grid

```
klo = int((m-x[0])*(npt-1)/ext);
```

(c) A spin-1 Zemach factor, as printed

$$\mathcal{M}_1 = s_{bc} - s_{ab}$$

A quiz. Which of these is wrong?

(a) A published fit-fraction budget

$$\sum_i \mathrm{FF}_i = 112.8\,\%$$

$$\sum_{i<j} \mathrm{FF}_{ij}\ \text{(printed)} = -6.33$$

(b) A spline-segment lookup, 41-knot grid

```
klo = int((m-x[0])*(npt-1)/ext);
```

(c) A spin-1 Zemach factor, as printed

$$\mathcal{M}_1 = s_{bc} - s_{ab}$$

All three. **And not one of them *looks* wrong on the page.**

Thesis-era C++ $\rightarrow$ 2026 Python

Then. Rio+, the Dalitz-plot fitter behind the LHCb $D^{\pm} \rightarrow \pi^{\pm}\pi^{\pm}\pi^{\mp}$ amplitude analyses. C++, shared folders, several hands. My PhD came out of it.

Now. Rebuilt in Python on the Scikit-HEP stack (`vector`, `particle`, `uproot`, `awkward`, `hist`, `mplhep`; `iminuit`; optional JAX) — *with LLM agents*.

3.9×10^{-15}

amplitudes vs the compiled binaries — parity, not tolerance

3.4×10^{-11}

Gauss–Legendre normalisation matrix vs the binary

435

machine-checked tests; fits close on toys from the *unmodified* legacy generator

What parity surfaced — mine first

Finding	Number	Adjudicated by
`find_bin` assumes a uniform knot grid; the thesis grid is not — the 2015 fit *likely* inherited it	wrong segment over 92 % of the mass range	TEST
Legacy quadrature order 100 leaves the S-wave normalisation entry off	5.3 %	CONVERGENCE SCAN
Cache check commented out in the shipped source; a naive re-enable mis-indexes	—	READING + CLOSURE
Published interference table omits the factor 2 of its own definition	-6.33 printed, -12.8 required	SUM RULE, BY HAND
$\omega(782)$ width under a [PDG2022] label matches neither edition	8.49 ± 0.13 printed; 8.68 ± 0.13	PDG LISTINGS
Spin-1 Zemach factor mistyped: $s_{bc} - s_{ab}$ for $s_{ac} - s_{bc}$	closure 998 % vs 2.4 %	PUBLISHED TABLE

Rebuilding surfaced these; the analyses stand — the Zemach factor is *mistyped*, the fit plainly used the standard one: that is why everything closes.

Thirty seconds: flip one convention

```
ZEMACH_CONVENTION = "standard"
rho(770)    25.48% 26.0±1.7     inside
omega(782)   0.10% 0.103±0.016  inside
rho(1450)    5.24% 5.4±1.6      inside
rho(1700)    5.60% 5.7±1.5      inside
f2(1270)    13.85% 13.8±0.49    inside
S-wave      62.87% 61.8±0.93    +1.2 sd
worst closure vs Table 2:  2.4%
$ pytest -q                13 passed in 0.96s
```

```
ZEMACH_CONVENTION = "as_printed"
rho(770)    11.02% 26.0±1.7     OUT −9 sd
omega(782)   0.04% 0.103±0.016  OUT −4 sd
rho(1450)    7.77% 5.4±1.6      +1.5 sd
rho(1700)   26.54% 5.7±1.5      OUT +14 sd
f2(1270)    24.02% 13.8±0.49    OUT +21 sd
S-wave      40.01% 61.8±0.93    OUT −23 sd
worst closure vs Table 2:  997.6%
$ pytest -q              3 failed, 10 passed
```

Terminal output, unedited (`capture/record.sh`): one `sed` on one flag between the panels.

Self-incrimination

Two weeks before this talk, a physical-region mask was written as

```
kibble(s12, s13, ...)  >= 0
```

— which selects the unphysical region.

- The code ran clean and passed every test that existed.
- It produced a *quantitative* diagnostic table.
- It yielded a confident conclusion — “the published Table 2 does not close” — which was written up and reported.

Self-incrimination

Two weeks before this talk, a physical-region mask was written as

```
kibble(s12, s13, ...)  >= 0
```

— which selects the unphysical region.

- The code ran clean and passed every test that existed.
- It produced a *quantitative* diagnostic table.
- It yielded a confident conclusion — “the published Table 2 does not close” — which was written up and reported.

What killed it: an identity, not a review

$\det(2\,p_i\cdot p_j) = -2\,s_{12}\,G \;\Rightarrow\; G \le 0$ inside — any masses, any point; exact to 3×10^{-12}.

Self-incrimination

Two weeks before this talk, a physical-region mask was written as

```
kibble(s12, s13, ...)  >= 0
```

— which selects the unphysical region.

- The code ran clean and passed every test that existed.
- It produced a *quantitative* diagnostic table.
- It yielded a confident conclusion — “the published Table 2 does not close” — which was written up and reported.

What killed it: an identity, not a review

$\det(2\,p_i\cdot p_j) = -2\,s_{12}\,G \;\Rightarrow\; G \le 0$ inside — any masses, any point; exact to 3×10^{-12}.

Where no oracle exists, this method produces confident, plausible, wrong work.

The inversion

The agents didn't make this code good. The oracles did.

And the oracles only got built because nobody trusted the agents.

What the record actually shows, read against the artifacts:

- **Every wrong number was adjudicated by an oracle** — a binary, a table, a closed form, an identity. None was settled by argument.
- **The reviewers' best catches were oracles run by hand**: a sum rule added up, two PDG editions fetched, a nesting bound $\mathrm{FCN}_{\text{nested}} \leq \mathrm{FCN}_{\text{parent}}$ applied. That is the case for making them *run*.
- **Reading caught what has no oracle**: spec drift against the C++, derivation errors in prose.

Convergence: the same wall, hit twice in one week

- **Same decay, same paper.** This week, at this workshop, ComPWA's maintainer (@rdeboer) opened a draft symbolic rebuild of $D^+ \to \pi^+\pi^+\pi^-$ from the same LHCb analysis.
- **Same wall, same fix.** The paper publishes fit fractions, not couplings — his magnitudes come from closure against the published fractions: the same closure that adjudicated the Zemach factor.
- **Same lesson.** His notebook works around a kinematics bug in his own package's form factor (s^2 for s; m_R for $\sqrt{s}$) — substituted, then *asserted*.

Independent stacks, one discipline: derive the number, then assert it.

ComPWA/report PR #44 (draft, 8 Sep 2026) · 2023, my own forgotten comment: "I refuse to think that it can't be done … Dpipipi".

Oracles, strongest first — and how much of my suite has one

Oracle	Why it can't be gamed	Caught here
IDENTITY	exact for any parameters; no tolerance to negotiate	mask sign; sum rule; nesting bound
BINARY	the old code has no idea it is being tested	3.9×10^{-15}; 3.4×10^{-11}
TABLE	someone else's numbers, fixed before you started	Zemach 998 % vs 2.4 %; ω width
CLOSED FORM	independent derivation, often much older	Parke–Taylor, known loop parts, ${}_2F_1$

Oracle coverage of my own 435 tests

42 % check a number against something independent (100 identity, 14 binary, 15 table, 55 closed form); 58 % do not: 17 % closures, 5 % self-pins, 35 % structural. Line coverage would tell you none of this.

Take-home: bring your legacy binary

1. **Dump** what the legacy side computes at fixed inputs: amplitudes, normalisation entries, PDF values — not summary plots.
2. **Pin the build**; **commit the record**; key it by inputs.
3. **Fail loudly** when the binary is unavailable — never skip silently.
4. Add the **identities** your code must satisfy: Bose symmetry, sum rules, nesting bounds, threshold behaviour. The only oracle theory code has.
5. **Your oracle's dependencies are not your library's dependencies**: ROOT validates the port and is never importable from it.

Harness and oracle inventory: intent to release. Today's transport strike has closed all my decay channels — I am completely stable at Nikhef.

Backup

Backup — who wrote the wrong number, and what settled it

	Author	Error	Settled by
1	thesis-era C++ (2015 rewrite)	uniform-grid `find_bin`, 92 % wrong segments	test (noticed by reading)
2	legacy config	GL order 100: $T[S, S]$ 5.3 % off	convergence scan
3	legacy source	cache check commented out; naive re-enable mis-indexes	reading + cold≡warm
4	LHCb 2023	interference table -6.33 vs -12.8	sum rule, by hand
5	LHCb 2022	ω width under wrong PDG label	PDG listings
6	LHCb 2023	Zemach $s_{bc} - s_{ab}$ mistyped	published fractions
7	agent, Aug 2026	mask >= 0: unphysical region	Gram-determinant identity
8	agent, Jul 2026	normalisation as $\mathrm{Re}(c^\dagger T c)$	sum rule at 3000 %
9	agent, Jul 2026	false minimum, $\mathrm{FCN}(6) > \mathrm{FCN}(5)$ nested	nesting bound, by hand
10	agent, Jul 2026	background normalisation acceptance-corrected	reading the C++
11	agent, Jul 2026	gluing sign, tadpole sign in prose	review (no oracle)
12	frontier LLMs	-46 for -23: doubled $2\,\mathrm{Re}$	the answer key…
13	the answer key	…inconsistent: exact integral gives -43	three-route closed form

Backup — oracle coverage, and what the audit turned up

class	*n*	**%**
identity	100	23.0
binary	14	3.2
table	15	3.4
closed form	55	12.6
closure	74	17.0
pinned	23	5.3
structural	154	35.4
total	435	

- The NLL is never compared to the binary — only its ingredients are.
- The benchmark pins are self-referential: the phase was inferred to reproduce the target.
- One discrepancy is encoded as a pass: an S-wave 1.2σ high, pinned to stay there.
- Identities have blind spots: every Zemach identity test is sign-blind; only the binary and the published 112.8 % see a flipped sign.

Backup — the oracle graded the grader

A benchmark item for frontier models: the ρ–ω interference ratio $r = I_{\rho\omega}/I_{\omega}$, answer $\mathrm{round}(10r)$.

- Intended key: narrow width, $r = -2.294 \Rightarrow -\mathbf{23}$. Common wrong answer -46: the $2\,\mathrm{Re}$ doubled.
- Exact integration of the *stated* integrals: -43 (infinite window), -41 (physical strip); and -46 is the $\Gamma_\omega \to 0$ limit of the exact residue — the "invalid" answer sat next to a legitimate one.
- Settled by three independent routes — closed form, residue + `mpmath`, finite window + `scipy` — the machinery that verifies the port.

Backup — same lesson from a second programme

Writing expert-physics questions to evaluate frontier models:

- half of what looked like model failure was *instrument* failure — a stale golden, an ambiguous prompt;
- in one case the golden was wrong and the model right eight times of eight;
- runs that *agree* on a non-golden value mean the golden is wrong; runs that *scatter* mean the prompt is under-specified.

A verifier that asserts the number certifies nothing. A 0/8 is necessary, never sufficient.