

PACE4 development and parallel interface update

Updated 27 October 2026 | Release 4.38.1 | Prepared for Taras

PACE4 4.38.1 corrects an out-of-bounds high-spin lookup that affected default calculations and produced FIND range errors. It retains the native parallel interface introduced in 4.38. This report documents the earlier physics comparison, parallel usage and performance, then the high-spin fix, its effect on default yields and its regression evidence. Affected calculations should be rerun; the earlier results used invalid table data.

Random sampling and reproducibility

On 26 October, initial spin selection was corrected to consume one random draw per cascade, outside the cumulative-distribution search. Energy loss retains its own draw. The masked Park-Miller generator is preserved. Default seeds use Qt system entropy and full-date milliseconds; PACE4_SEED accepts integers from 1 to 2147483646. The external seed is recorded in HTML and RTF reports.

Input save and load corrections

Compound LMINN is now read from its sixth field instead of duplicating AJNUC. Floating parameters are saved with 17 significant digits and explicit separators. Loaded excitation energy and barrier settings are refreshed before execution. Imported spin weights are validated for counts, finite nonnegative values, and usable totals; missing legacy tail weights are cleared and the full spin range is saved.

Validation and resource cleanup

Bounds checks cover compound spin, editor paging, cascade count, minimum partial waves, and mode flags. Invalid batch counts and ranges are rejected before calculating step energies; batches stop on failure or cancellation. Vector ownership and scoped output cleanup replace unsafe lifetime handling. Stream failures and composition errors propagate to silent-mode status.

Release and project maintenance

ftype.h was renamed to g_PACE/pace_version.h, with includes and PACE.pro updated. The application and executable metadata now use version 4.38.1, dated 27 October 2026. The header records the earlier physics fixes, native parallel interface and high-spin correction. The stylesheet loading correction and test/ ignore rule remain in place. This report is maintained in g_PACE/doc/.

Confirmed physics corrections and tests

Continue the charged particle search through empty bins

CHNPRB in PACE3Levden.cpp previously returned when an upper residual-energy bin produced no probability entries. Lower bins can still contain open proton or alpha transitions. The corresponding return is commented out in the original PACE3 FORTRAN. The active C++ return and its unused bookkeeping were removed; all residual-energy bins are now searched.

The regression constructs an empty upper bin followed by a populated lower bin with an open proton optical channel. The earlier code discarded the transition; the corrected code finds the expected lower-bin state. The production benchmark now contains proton and alpha residues.

Preserve half integer compound spin

CHNPRB stored $AJC = c.J - 1 + SC$ in an integer. For an odd-mass nucleus, $SC = 0.5$, so this truncated the physical spin and changed angular-momentum selection. AJC is now double, consistent with the FORTRAN real-valued variable. The regression verifies an allowed $J = 1/2$ to $J = 1/2$ E1 transition and confirms that $J = 0$ to $J = 0$ remains forbidden.

Initialize every rotational energy for nonzero FYRST

GCLDP in PACE3compound.cpp previously used break after initializing the first rotational-energy entry when FYRST was nonzero. It now fills every required spin entry. The regression checks all entries across repeated calls with FYRST = 1, 0.625, and 0, exposing missing or stale values in the earlier source.

Restore the interior Coulomb potential

PACE3util.cpp divided the entire expression $(3 - r^2)$ by RC^2 . The original FORTRAN expression is ZZI times $(3 - r^2 / RC^2)$. The corrected parentheses restore the uniformly charged sphere potential and continuity with the exterior potential at the nuclear surface. Proton and alpha optical coefficients are checked against values generated from the original FORTRAN routines.

Verification	Result
All four targeted tests against earlier physics sources	Fail as expected
Targeted physics tests and optical coefficients	Pass
MSVC and MinGW full application builds	Pass
MSVC versus MinGW residue counts across 30 runs	Identical
Nine silent-mode and error-path regression cases	Pass
Qt load/save modes 1 2 3 5 and cleanup tests	Pass
RNG replay and one-draw spin sampling tests	Pass

The tests exercise production physics functions. Before-and-after evidence and reference optical coefficients are retained under `_build_validation/`. The original FORTRAN files were not edited.

Benchmark method and comparison results

Ni-60 at excitation 30.125 MeV and zero recoil was evaluated with fixed $J = 10$, equal weights at $J = 0, 10, 20$, and generated weights proportional to $2J + 1$ for $J = 0$ through 20. All three cases were run at $\text{FYRST} = 0$ and $\text{FYRST} = 1$. Five replicas of 5,000 cascades give 25,000 events per case and 150,000 events per implementation across six cases.

Other settings were $\text{BARFAC} = -1$, $\text{ARATIO} = 1$, $\text{FACLA} = 10$, $\text{NOSHL} = 1$, $\text{IDIST} = 0$. Input normalization was 100 mb for fixed and equal-weight cases and 441 mb for generated weights. These are conditional compound-decay cross sections under supplied normalization, not independent entrance-channel fusion predictions.

PACE4 seeds were 436, 104729, 130363, 155921, and 196613. The primary FORTRAN reference used these values plus 1000003. Original inputs, seeds, executable hashes, and outputs are retained in the run manifests. Each MSVC result matches its corresponding instrumented MinGW result exactly.

Initial spin sampling remains consistent

Distribution	PACE4 fit p	FORTTRAN fit p	Comparison p
Equal $J = 0, 10, 20$	0.9632	0.0951	0.3442
$2J + 1$ for $J = 0$ to 20	0.4942	0.6483	0.4725

These initial-spin statistics are identical at $\text{FYRST} = 0$ and 1; every fixed-spin event has $J = 10$. They do not establish agreement of all spin observables during the decay chain.

Residual distribution comparisons

FYRST	Spin input	Chi square	df	p
0	Fixed $J = 10$	12.98	6	4.33e-02
0	Equal $J = 0, 10, 20$	2.63	7	9.17e-01
0	$2J + 1$ for $J = 0$ to 20	7.43	7	3.86e-01
1	Fixed $J = 10$	12.48	7	8.59e-02
1	Equal $J = 0, 10, 20$	14.43	7	4.40e-02
1	$2J + 1$ for $J = 0$ to 20	1.83	7	9.68e-01

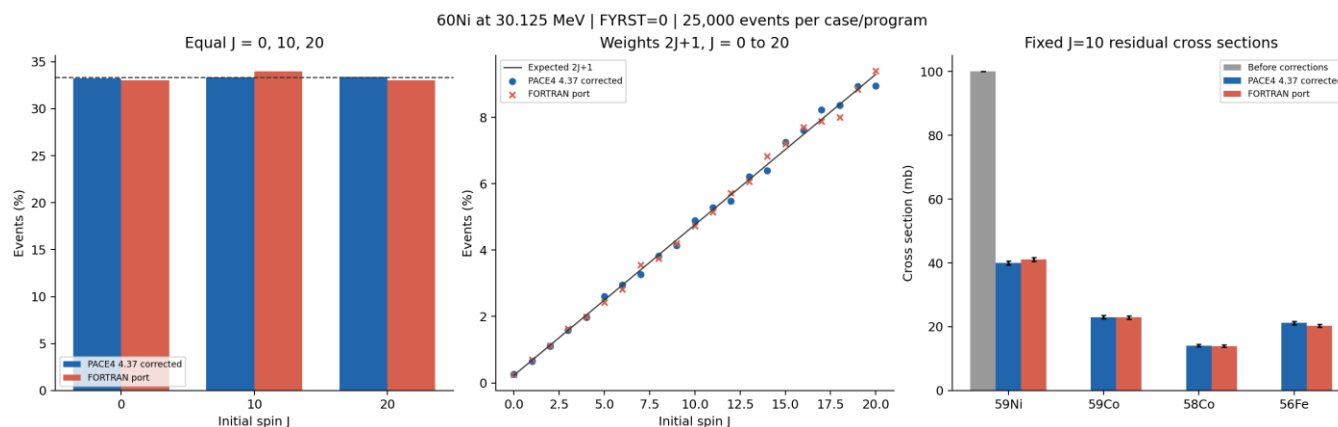
None of the six residual comparisons rejects agreement at the familywise 5 percent threshold. This is evidence of agreement within the tested scope, not general physics validation. Two cases remain marginal using an unadjusted 5 percent threshold: fixed $J = 10$ at $\text{FYRST} = 0$ ($p = 0.0433$), and equal weights at $\text{FYRST} = 1$ ($p = 0.0440$). Larger independent samples and broader nuclear cases are warranted before claiming general validation. Rare categories are pooled where expected counts permit; excluded remainders are listed in `summary.json`, with full counts in the CSV. Statistical intervals exclude model and portability uncertainty.

Cross section improvements and numerical checks

Fixed $J = 10$, FYRST = 0, 25,000 cascades, and input normalization 100 mb. All values below are in mb. The earlier release column is the baseline before the four physics corrections.

Residue	Earlier release	Corrected PACE4	FORTTRAN reference
59Ni	100.000	40.016	41.048
59Co	0.000	23.000	22.928
58Co	0.000	14.108	13.956
56Fe	0.000	21.192	20.304
55Fe	0.000	1.572	1.600
58Ni	0.000	0.012	0.044

Ni-59 has a corrected 95 percent Wilson interval of 39.410 to 40.625 mb; the reference interval is 40.440 to 41.659 mb. The fixed-spin residue comparison changed from chi-square about 20,894 before corrections to 12.982, with $p = 0.04332$.



Gray bars show the earlier release, blue the corrected build, and red the portable FORTRAN reference. Error bars on cross sections are individual 95 percent Wilson intervals.

Diagnostics explain why each correction was needed

Removing only the premature return restored charged-particle channels but left large differences. Preserving half-integer spin improved daughter-decay selection. The three-fix build still had a first-decay proton weight about 11 percent below the reference, which led ASTRA to identify the Coulomb-potential error. Intermediate results are preserved to separate these effects.

The positive C++ probability cutoff was retained. An isolated reference experiment correcting historical logarithmic-threshold behavior shifted fixed-case Ni-59 counts only from 10,262 to 10,238. PACE2A.FOR also supports the positive product rule. Numerical precision, cutoff behavior, and Monte Carlo variation should be considered when interpreting the comparison; exact trajectory identity is not claimed.

After the Coulomb correction, the first-decay proton raw weight is $1.47301e-34$ versus $1.47666e-34$ in the reference, and α is $1.02773e-34$ versus $1.02779e-34$. Differences in retained entry counts remain; matching channel totals and final distributions does not imply identical internal tables.

Reference limitations and saved evidence

What the FORTRAN reference represents

The reference is a documented GNU port of PACE3.for, not a pristine historical VAX executable. Twelve VAX OPTIONS /G_FLOATING directives were commented out. Compiler flags include -std=legacy, -fdec, -fallow-argument-mismatch, -ffixed-line-length-72, -fno-automatic, -O0, and -fcheck=bounds. Static local storage preserves historical tables across calls.

GNU RAN(seed) returned 0.041216135 on five consecutive calls without updating the seed. The isolated port uses a portable masked Park-Miller recurrence with REAL*4 output, retaining the original 30 warm-up draws and energy-before-spin draw order. Snapshot instrumentation consumes no additional draws. Default REAL*4 and explicit REAL*8 declarations are retained; the C++ implementation uses mostly double precision.

A guard avoids a zero-weight out-of-range neighbor at the FISROT IY = 11 boundary. In the earlier FYRST = 0 validation, all 15 checked/repared and unchecked/unrepaired runs had identical spin and residue counts. The current reruns use the checked reference. COMMON-block size warnings remain a portability limitation. PACE2A.FOR was not used as the main executable reference.

Reproducible results

Project root: C:\Users\taras\Documents\Codex\PACE4

Current results: _build_validation/physics_4_37_corrected/

report/report.html, report/summary.json, report/cross_sections.csv - FYRST = 0 results and before-and-after comparison

fyrst1/report/ - FYRST = 1 comparison

cpp_msvc/, cpp_mingw/, fortran_independent/ - primary manifests and per-run outputs; the same structure appears under fyrst1/

first_two_fixes/ - intermediate MSVC results before the half-integer correction

before_coulomb_fix/ - three-fix manifests, raw results, and report; also retained under fyrst1/

cpp_trace/first_decay.csv - corrected channel-level trace

fortran_positive_threshold/ - isolated reference cutoff experiment

The earlier baseline remains in _build_validation/physics_4_37/. Reusable drivers are tests/physics_benchmark.py, tests/fortran_benchmark.py, and tests/compare_physics_benchmarks.py. Compiled targeted tests are tests/physics_regression.cpp and tests/physics_regression.pro. The C++ runner now accepts --fyrst for matched parameter comparisons.

tests/generate_optical_reference.py extracts the original optical routines and records source hashes and compiler flags. Coefficients and provenance are saved in optical_reference/. The optical regression tolerance is 1e-6 absolute; observed maximum differences were 2.44e-15 for neutrons, 3.45e-10 for protons, and 4.23e-8 for alpha particles. The earlier proton error reached 0.323.

Original PACE3.for SHA256 remains unchanged:

c4149ec9ff1a6ebbd94ff0d1199fb925a3720636e97e1a0de2a19c8f77aeec2f

Independent processes and further improvements

Implemented independent calculation processes

Implemented with ASTRA: tools/pace_parallel.py schedules a bounded number of silent PACE4 processes, one per complete input or replica. Each job receives a private runtime, mass database, output folder and traceback file. Explicit seeds are tied to stable job IDs. Four workers achieved a measured median speedup of 2.56 times; the next page gives the method and results.

Preserve the numerical probability cutoff

PACE3main.cpp sets $\text{DISCPR} = 1 / \max(500, 3 \sqrt{\text{NCASC}} + 10)$. The runner copies each input byte-for-byte and never divides or overrides NCASC. Whole jobs therefore keep the original cutoff. At 40,000 cascades it remains $1/610$, approximately 0.0016393443, in every tested schedule. Splitting a single calculation into smaller runs is intentionally unsupported because it can change the cutoff.

Isolate simulation state before adding threads

The current engine has mutable global probability and level-density arrays, static CHNPRB tables, a static RNG state, global file handles, Qt output objects, and a shared database connection. Adding an OpenMP loop around cascades would risk races. A per-worker simulation context should own mutable state and RNG streams, while validated immutable tables can be shared. GUI interactions should stay on the GUI thread and worker results should be combined deterministically.

For reproducibility across scheduling changes, assign streams to stable calculation or event identifiers, not whichever worker happens to execute next. Check stream separation and avoid duplicate seeds. Merely making the RNG thread-safe would not make the entire simulation engine thread-safe.

Prioritize correctness and measured bottlenecks

First compare channel probabilities and selected transitions with the reference for the same nuclear state, then expand to other nuclei, excitation energies, and model options. Add invariants for nonnegative probabilities, allowed angular momentum, valid table indices, and event-count conservation. Record source revision, executable hash, seed, model options, and effective cutoff in every benchmark manifest.

Profile computation separately from HTML/RTF and particle-output generation before optimizing. Candidate improvements include caching verified immutable transmission or level-density data and offering a lightweight results mode for large parameter scans. Each optimization should retain deterministic regression coverage and statistical comparisons.

Technical references

Qt documentation on reentrancy and thread safety: <https://doc.qt.io/qt-6/threads-reentrancy.html>

OpenMP 5.2 threadprivate rules: <https://www.openmp.org/spec-html/5.2/openmpse24.html>

The external process runner and native Qt parallel interface are implemented and tested. Engine threading and the additional optimizations above remain future work. Release 4.38.0 adds the interface; no simulation algorithm was changed for parallel execution. The earlier FORTRAN comparison and timing measurements were performed with corrected 4.37 builds.

Parallel execution performance and verification

The same 12 complete calculations were run with one, two and four workers in three counterbalanced rounds. Each run contained 480,000 cascades, totaling 4.32 million across nine runs. The workload used Ni-60 at 30.125 MeV, spin modes 2, 3 and 5, FYRST 0 and 1, and two fixed-seed replicas per case. Each job retained 40,000 cascades.

Workers	Median seconds	Range seconds	Speedup
1	27.54	25.08 to 42.09	1.00 times
2	16.66	15.68 to 18.43	1.65 times
4	10.74	10.55 to 11.07	2.56 times

Measured on an Intel Core i7-1255U with 10 physical cores, 12 logical processors and 32 GB RAM, using the corrected MSVC deployment. End-to-end wall times include Python startup, runtime copies, application startup, calculation and output. The one-worker range shows substantial variation; these three-sample medians describe this machine and workload, not a guaranteed speedup. These timings measure the command-line process runner on 4.37; the new GUI scheduler was verified for correctness but was not separately speed-benchmarked.

Reproducibility and failure handling

All 108 completed jobs conserved event counts. Each job had identical input bytes, seed, cutoff, CS4 hash and residue counts across schedules. Ten runner regression tests passed separately with MSVC and MinGW. Real IDIST 0, 1 and 2 cases also reproduced particle and binary traceback output exactly between serial and parallel runs. HTML and RTF include dates and paths, so complete report bytes are not expected to match.

Tests cover invalid manifests and seeds, existing output folders, missing data, missing output, launch failures, per-job timeouts and interruption. A failure stops active children and cancels pending jobs. Logs and results.json retain status, hashes, seeds and timings. Duplicate effective RNG starting states are rejected; distinct states alone do not prove nonoverlapping streams of arbitrary length.

Running the included example

Python 3.9 or newer is required. Run from the project root with the installed PACE4 libraries and plugins present. Select a new output directory for each batch:

```
python tools/pace_parallel.py --exe _install_MSVC/PACE4.exe
--manifest tools/examples/parallel/jobs.json
--out test/parallel_example --jobs 4 --timeout 180
```

Enter the three displayed lines as one command. The manifest assigns each job an ID, input path and explicit seed. Use --jobs 1 for a serial reference. Detailed usage and isolation rules are in tools/README_parallel.md. Results from different physical inputs must not be pooled as interchangeable replicas.

Saved validation evidence

Timing samples, hashes and per-job results: _build_validation/parallel_speed_4_37/timing.json and its run folders. Regression logs: _build_validation/parallel_regression_msvc.log and parallel_regression_mingw.log. Reusable tests: tests/parallel_regression.py and tests/parallel_benchmark.py.

Using the native parallel interface

Open File > Parallel calculations..., click Parallel... on the toolbar, or press Ctrl+Shift+P. The dialog runs complete, independent calculations through Qt child processes. Python is not required. Save edits in the main window first: the queue reads saved files, and an existing current input is added automatically.

1. Click Add inputs... and select one or more saved .pace files. Use Duplicate selected to create another replica, or Remove selected to remove a row.
2. Review the cascade counts and editable seeds. Each row needs a distinct effective RNG starting state. Keep its input and seed unchanged to reproduce its scientific output.
3. Set Workers to the maximum number of simultaneous calculations. The range is 1 to 64; the initial value is 2 when at least two logical processors are available. One worker provides a serial reference. Higher counts require more memory.
4. Choose a new Results folder. Browse selects a parent and proposes a new batch subfolder. Existing folders are refused. Optionally set Timeout per job in seconds; No limit disables the timeout.
5. Click Start batch. Watch per-job status, completed-job progress and the log. Cancel batch stops active children and cancels queued jobs; closing the dialog during a run also cancels and waits for cleanup.
6. Select a completed row and click Open results to open its HTML report. With no row selected, this opens the batch folder. Run again retains row identities and seeds while proposing a fresh folder; input files are read again, so preserve them for replay.

Add saved .pace files. Duplicate an input to run another replica with a different seed. Save changes in the main window first. Each calculation keeps its original cascade count.

Add inputs... Duplicate selected Remove selected

	Input file	Cascades	Seed	Status
1	source-1.pace	128	104729 ^ v	pending
2	source-2.pace	128	235092 ^ v	pending

Workers: 2 ^ v Timeout per job: No limit ^ v

Results folder: C:/Users/taras/AppData/Local/Temp/parallel_dialog_regression-cyEzTt/result-3 Browse...

Choose a new folder for this batch. Existing folders are kept unchanged.

0%

Ready. Add one or more input files.

Open results Start batch Cancel batch Close

Verified dialog layout with two test inputs. Workers controls concurrency only; it does not divide a calculation or reduce its cascade count.

Native interface validation and release evidence

Reproducibility and probability cutoff

The GUI copies each input byte-for-byte and keeps NCASC unchanged. Its private per-job executable, mass database, configuration and traceback folder prevent shared-file collisions. Seeds are assigned to stable jobs, independently of worker scheduling. Duplicate effective masked RNG states are rejected. Distinct states alone do not prove nonoverlapping streams of arbitrary length.

The real serial-versus-parallel test uses a 40,000-cascade case, verifying DISCPR = 1/610, approximately 0.0016393443, and unchanged input bytes. Additional IDIST 1 and 2 cases verify nonempty particle and binary traceback output. CS4, particle and traceback bytes match exactly between serial and two-worker runs. HTML and RTF contain dates and paths and are not expected to match byte-for-byte.

Completed checks

Check	Result
Native Qt suite including setup and cleanup	12 passed, 0 failed
Worker limit, seed replay and masked-seed alias rejection	Passed
Existing folders, missing outputs and launch errors	Passed
Failure cancellation, timeout and window-close cleanup	Passed
Injected results.json write failure and terminal status	Passed
Main-window toolbar action and modal worker setting	Passed
Existing input load/save/reload and cleanup tests	Passed
Real scientific replay with installed MSVC and MinGW builds	Passed
Version 4.38 application builds and release metadata	Verified

The tests exposed clipped numeric controls and stale terminal summaries after a persistence failure. Both were corrected. The write-failure regression fails before the fix and passes afterward; a batch now stops on persistence failure and retains accurate terminal states in memory. The exact trigger of one earlier intermittent I/O failure was not established; repeated real calculations subsequently passed.

Evidence and scope

Native implementation: g_PACE/parallel_dialog.cpp and .h. Usage: tools/README_parallel.md. Tests: tests/parallel_dialog_regression.cpp and .pro; main-window integration is included in tests/input_roundtrip.cpp.

Saved logs are under _build_validation/parallel_dialog_tests/: results.txt, real-repeat.txt, installed-msvc-results.txt and installed-mingw-results.txt document the interface validation. Release 4.38 checks are recorded in release438-msvc.txt and release438-mingw.txt. Main-window integration evidence is in _build_validation/roundtrip/gui_integration_test_windows.log.

The earlier FORTRAN cross-section and spin-distribution evidence on pages 2 to 5 is retained with its original statistical and portability limitations. The 2.56-times four-worker speedup on page 7 belongs to the earlier process-runner benchmark; it is not a new GUI timing claim.

High spin correction and default calculation results

Cause of the default FIND errors

The default Ca-48 plus Sn-124 calculation at 300 MeV reaches initial spins around 148. Fission-barrier storage and initialization still used a legacy 110-entry spin range, while the application accepted spin indices through Max_MOM = 200. A high-spin barrier lookup could therefore read unrelated memory. In the reproduced MinGW run, this generated requests at 216.279 and 202.331 MeV against a level-density table ending at 144.5 MeV.

The 216.279-MeV request is consistent with the compound excitation 143.076 MeV minus a bogus negative barrier near the reaction Q-value of -73.203 MeV. The diagnostic also used the neutron-channel label Z = 70, N = 101 for the fission lookup. The Gilbert–Cameron spin-cutoff message itself is informational; the FIND warnings were a code defect, not an instruction to alter the default input.

What changed

Fission-barrier storage, initialization, level-density spin loops and scratch arrays now consistently use Max_MOM. Level-density capacity is checked before writing a row. FIND stops at the final valid energy bin, including an exact endpoint request. FISROT skips a nonexistent next-row read when its interpolation weight is zero. Original inputs, cascade counts, RNG policy and probability cutoff are retained.

Effect on the reproduced default calculation

Fixed seed 436; 1,000 cascades; Ca-48 + Sn-124 at 300 MeV; FYRST = 0; FACLA = 10; NOSHL = 0; classical entrance barrier; unchanged saved optical inputs. The compound nucleus is Yb-172 at 143.076 MeV excitation. The supplied run normalization is approximately 1,580.45 mb. Both columns use the same input and seed; values are simulation outputs, not measurements.

Observable	Before fix	Corrected
Displayed FIND range warnings	10 then suppressed	0
Fission events	248 of 1000	299 of 1000
Fission fraction	24.8%	29.9%
Fission cross section in mb	391.951	472.554
Yb-172 residual events	270 of 1000	224 of 1000
Yb-172 cross section in mb rounded	427	354
Cascades reported as spin inhibited	35.2%	30.7%
Total accounted events	1000	1000

Fission increased by 51 events, or 5.1 percentage points; Yb-172 decreased by 46 events. The correction changes decay probabilities and subsequent random trajectories. This comparison demonstrates the practical effect of the defect; it is not a statistical comparison with experimental data.

High spin validation and remaining limits

Targeted tests distinguish the corrected behavior

Targeted check	Earlier sources	Corrected
All supported fission-barrier entries initialized	Fail	Pass
Level densities include the supported spin range	Fail	Pass
Capacity checked before writing the level vector	Fail	Pass
FIND stays within its last valid bin	Fail	Pass
Fission interpolation at the final table row	Pass numerically	Pass
Earlier rotational energy and repeated-call test	Pass	Pass
Half-integer dipole selection test	Pass	Pass
Optical coefficients against original FORTRAN	Pass	Pass
Charged channel after an empty energy bin	Pass	Pass

All nine targeted tests pass. Four fail against saved pre-fix sources, using expanded barrier allocation to avoid a test-side overrun. The old interpolation can pass numerically despite an invalid zero-weight read; that fix follows from array-index inspection.

End to end and release checks

Seven cases pass on MSVC and MinGW: default FYRST = 0 and 1 with two seeds each, plus three Ni-60 low-spin distributions. All conserve event totals without FIND or capacity warnings. CS4 hashes and residue counts agree exactly across compilers. The three low-spin controls retain their pre-fix CS4 bytes. Both installed 4.38.1 builds were checked again.

Remaining interpretation limits

Spin-inhibition traps remain. The FYRST = 0 Gilbert–Cameron rotational-energy model can exclude high-spin states admitted by the entrance-channel yrast calculation. This is distinct from invalid memory access. The remaining 30.7 percent must not be described as a solved artifact or proof of general physics validation.

The original FORTRAN retains a historical 110-spin limit; this fix makes C++ tables consistent with its existing Max_MOM = 200 interface. Earlier low-spin comparisons remain valid within their stated scope. No new high-spin FORTRAN agreement or experimental validation is claimed. Rerun affected calculations.

Saved evidence

Evidence root: `_build_validation/default_find/`. `before_mingw/` and `after_mingw/` retain the exact reproduction. `verified_msvc/` and `verified_mingw/` hold seven-case outputs and `scientific_results.json`. `low_control_before/` holds the earlier low-spin controls; `regression_before/results.log` records the four expected failures.

Targeted log: `_build_validation/physics_regression/default_after_results.log`. Reusable check: `tests/default_physics_regression.py`. Release reruns: `default_find/release4381_msvc/` and `release4381_mingw/`. Changelog: `g_PACE/pace_version.h`.

Parallel interface help and verification

Release 4.38.1 now includes a Help button at the bottom left of the Parallel calculations dialog. Press F1 while that dialog is active to open the same help window. The resizable window provides selectable, scrollable instructions and can remain open during a batch. Closing Help does not cancel calculations.

The guide covers adding and duplicating saved inputs, reproducible seeds, worker counts, new results folders, timeouts, cancellation, opening reports and replay. It explains that every job keeps its complete cascade count and probability cutoff; workers run separate calculations concurrently.

Using the native parallel interface

Open **File > Parallel calculations...**, click **Parallel...** on the toolbar, or press **Ctrl+Shift+P**. The dialog runs complete, independent calculations through Qt child processes. Python is not required. Save edits in the main window first: the queue reads saved files, and an existing current input is added automatically.

1. Click **Add inputs...** and select one or more saved .pace files. Use **Duplicate selected** to create another replica, or **Remove selected** to remove a row.
2. Review the cascade counts and editable **seeds**. Each row needs a distinct effective RNG starting state. Keep its input and seed unchanged to reproduce its scientific output.
3. Set **Workers** to the maximum number of simultaneous calculations. The range is 1 to 64; the initial value is 2 when at least two logical processors are available. One worker provides a serial reference. Higher counts require more memory.
4. Choose a new **Results folder**. **Browse** selects a parent and proposes a new batch subfolder. Existing folders are refused. Optionally set **Timeout per job** in seconds; **No limit** disables the timeout.
5. Click **Start batch**. Watch per-job status, completed-job progress and the log. **Cancel batch** stops active children and cancels queued jobs; closing the calculations dialog during a run also cancels and waits for cleanup.
6. Select a completed row and click **Open results** to open its HTML report. With no row selected, this opens the batch folder. **Run again** retains row identities and seeds while proposing a fresh folder; input files are read again, so preserve them for replay.

Each job keeps its complete cascade count and probability cutoff. Workers controls how many separate jobs run at once; it does not split one calculation. Changing the worker count does not change a job's seed. Different physical inputs must not be pooled as replicas.

You can leave this help open during a batch. Closing this help window does not cancel calculations.

Close

Checks after adding Help

Check	Result
Msvc and MinGW builds and installed executables	Updated successfully
Existing parallel dialog suite	11 passed, 0 failed, 1 skipped
Windows Help opening, closing and reopening smoke check	Passed
Windows screenshots of main dialog and Help	Visually checked

The smoke run reports three passes including setup and cleanup. It checks help visibility and content, reopening, and that closing Help leaves the parent dialog open. F1 binding was checked in source. The optional real-physics replay was not enabled in this UI-only rerun; earlier physics results remain on pages 9 to 11.

Evidence: `_build_validation/parallel_dialog_tests/help-regression.txt` and `_build_validation/help_smoke/results-windows.txt`. Help implementation and F1 binding: `g_PACE/parallel_dialog.cpp`. Version remains 4.38.1.