TECHNICAL MODIFICATION GUIDE

LISE++ 18.7 Secondary-Reaction AA Modifications

Direct AA routing, Last-Z behavior, optional intermediate output, cache classification, equivalence diagnostics, and a safe path to parallel execution

Versions 18.7.0 through 18.7.7

Revision date: August 27, 2026

Oleg B. Tarasov

tarasov@frib.msu.edu

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics and used resources of the Facility for Rare Isotope Beams (FRIB) Operations, which is a DOE Office of Science User Facility under Award Number DE-SC0023633, and by the US National Science Foundation under Grants No. PHY-20-12040 and 23-10078 “Windows on the Universe: Open Quantum Systems in Atomic Nuclei at FRIB”.

1. Purpose and Executive Summary

The 18.7 development line addresses Secondary Reactions when Abrasion-Ablation (AA) is selected for the second reaction. The motivating 144Sm to 108Xe case produced an apparent secondary enhancement near 10^11. Investigation showed that calculation control and observability were as important as the cross-section model itself: Secondary Reactions had overridden the user AA mode, the Last-Z selection could expand unexpectedly, and earlier diagnostics did not distinguish a reused parent array from a cached scalar pair or an absent daughter.

The current 18.7.7 implementation preserves the user's AA FAST choice, accepts the selected Secondary Last-Z value in the agreed timeout and silent modes, uses the established direct EPAX::yieldSecondaryAA calculation path, optionally writes the intermediate parent-to-residue matrix, and reports separate SR-map, AA-parent, and daughter classifications.

Important historical correction The persistent Base_SR scalar cache introduced experimentally in 18.7.1 was constrained in 18.7.4 and removed in 18.7.5. It is not part of the current 18.7.7 production path. The run-local vmpCS map and AA's current parent manager remain active and are now reported separately.

2. Release History Since 18.7.0

Version	Date	Status	Principal change
18.7.0	Aug. 22	Baseline	Opened the 18.7.x development line.
18.7.1	Aug. 26	Historical experiment	Removed forced AA FAST mode; introduced persistent scalar caching and initial cache diagnostics.
18.7.2	Aug. 26	Retained	Added plot-subtitle markers for calculated versus User-Calc/file first-step cross sections.
18.7.3	Aug. 26	Retained	Kept the interactive timeout; Secondary AA timeout and silent/fit modes accept the selected Last-Z value.
18.7.4	Aug. 26	Historical transition	Restricted scalar caching to one SR calculation and tightened parent/range verification.
18.7.5	Aug. 26	Current foundation	Removed the experimental scalar-cache and wrapper path; restored direct yieldSecondaryAA routing.
18.7.6	Aug. 26	Current	Added an optional AA intermediate-output checkbox and contextual file header.
18.7.7	Aug. 27	Current	Separated SR-map and AA-parent reuse; classified daughters; added one direct equivalence probe.

3. Motivation and Control Fixes

3.1 User-selected AA mode

Secondary-Reaction initialization no longer forces opt->AA_fast=1. FAST use and AA_fastMode now come from the same user settings used by a direct projectile-fragmentation calculation. Therefore, NPevap and evaporation-velocity behavior follow the normal AA rules rather than an SR-only override.

3.2 Secondary Last-Z behavior

Context	Current behavior
Interactive SR with AA	The 3-second timeout remains. Timeout accepts the currently selected Last-Z mode/value.
Silent or fit mode	The current selection is accepted immediately; lastZ_Mode is not forced to All.
Non-AA secondary model	The previous non-AA timeout behavior remains unchanged.
After calculation	G_LastZ_ask, G_LastZ_value, and G_LastZevaporation are restored.

The Last-Z value controls how far each AA evaporation tree is generated. A value that is too low increases work and memory; a value that is too high can make requested lower-Z daughters absent. The selected boundary must therefore cover every final charge that the SR filter will request.

4. Current Secondary-AA Architecture (18.7.5-18.7.7)

4.1 Direct projectile substitution

EPAX::yieldSecondaryAA saves the global beam and reaction, substitutes the intermediate SR parent as the projectile, sets fragmentation mode, calls the established AA yield function, and restores the original state. This is the production route restored in 18.7.5.

```
SR parent (Zp,Np) -> EPAX::yieldSecondaryAA(parent, daughter, energy)
  save beam + reaction; set parent as beam; calculate AA daughter
  inspect AA manager for diagnostic status; restore beam + reaction
```

4.2 Run-local SR map

Each SR invocation owns vmpCS[parentIndex], a sparse map from daughter isotope index to scalar cross section. A map miss calls AA once and inserts the returned value. Later target-thickness steps use the stored value. This map is destroyed at the end of the SR calculation; it is not a persistent cross-run cache.

4.3 AA parent manager

Within the direct AA layer, one ManagementAA object can supply many daughters for the same parent. It is reusable only when the current manager and AAcompare snapshot match the substituted parent, target, energy-dependent settings, AA options, and the requested Last-Z range. Switching to another parent normally constructs another AA manager.

4.4 Why scalar persistence was removed

The 18.7.1 persistent scalar cache attempted to preserve individual parent-daughter values beyond one SR invocation. Runtime tracing showed that this experimental wrapper could obscure which parent AA array had actually been generated. Version 18.7.5 deliberately returned to the established direct route. Current optimization should build on parent-level work packages with explicit state ownership, not reintroduce an implicit global scalar cache.

5. Version 18.7.7 Diagnostic Model

The current log separates three layers that were previously compressed into one line.

Layer	Counter	Exact meaning
SR vmpCS	SR map hits	A parent-daughter scalar was already stored during this SR invocation.
SR vmpCS	SR map misses	The current invocation had not requested this pair; yieldSecondaryAA was called.
AA parent	hits	A compatible parent manager already existed and covered the requested daughter charge.
AA parent	parent calculations	No compatible parent/range was reusable before the request; a parent calculation was required.
AA daughter	positive	The daughter node existed and its raw CCSrest was greater than zero.
AA daughter	physical zeros	The daughter node existed, but its raw CCSrest was exactly zero.
AA daughter	absent	ManagementAA contained no node for the requested daughter.
AA daughter	unphysical	The raw daughter value was negative or non-finite.

Do not merge zero and absent EPAX may floor a non-positive AA result to a tiny value such as 1e-99 at the public yield layer. Version 18.7.7 inspects ManagementAA::TryGetCS so the diagnostic can distinguish an existing zero daughter from a daughter that was never generated.

5.1 Current output form

```
SR AA routing total active parent visits <n> SR map hits <n> SR map misses <n>
Secondary AA parent cache: <n> hits, <n> parent calculations
Secondary AA daughters: <n> positive, <n> physical zeros, <n> absent, <n> unphysical cross
sections
Secondary AA single-step equivalence: parent Z N fragment Z N SR <x> direct <y> relative
difference <r> PASS|FAIL
```

5.2 Interpreting the historical 17,415-request example

The earlier line '0 hits, 17415 misses, 252 parent calculations, 0 unphysical cross sections' belonged to the experimental 18.7.1-18.7.4 scalar-cache instrumentation. It should not be expected verbatim in 18.7.7. Its useful computational conclusion remains: 17,415 scalar requests were served by roughly 252 expensive parent calculations, or about 69 requested daughters per parent calculation.

Derived quantity	Value	Interpretation
Requests per parent calculation	17,415 / 252 = 69.11	Parent-level work is the dominant optimization target.
Requests reusing a parent	17,163 (98.55%)	Most daughter extraction reused the current parent array.
Likely 18.7.7 mapping	SR misses / parent hits / calculations	Exact values must be read from a new 18.7.7 run.

6. Single-Step Equivalence Diagnostic

For the first requested daughter that exists with a positive raw AA cross section, SR stores the production value and repeats exactly the same direct yieldSecondaryAA request once. The repeated value is used only for comparison; it never replaces the production value in vmpCS.

The relative difference is |SR-direct| / max(|SR|, |direct|). PASS requires finite values and a relative difference no larger than 1e-12. The check verifies repeatability of the direct request and parent-cache path. It is not an independent physics-model validation and does not compare FAST against a higher-NPevap reference.

Failure response A FAIL indicates state leakage, unstable cache reuse, or non-determinism and should block any parallel implementation until resolved.

7. Optional SR_AA_intermediate_v1.txt Output

Version 18.7.6 adds 'Write AA intermediate cross sections' to the Secondary Reactions dialog. The checkbox is off by default, enabled only when the secondary model is AA, and stored in both user options and LISE project files. The user's current layout displays the checkbox on two lines beside the secondary cross-section selector.

When enabled, successful yield and kinematics calculations write SR_AA_intermediate_v1.txt in the results path. Cancellation or failure does not produce a new file. The header records calculation mode, number of steps, participants, parents, links, invalid values, energy, AA FAST settings, and target.

```
! calculation_mode=yield|kinematics steps=<n> participants=<n> parents=<n> links=<n>
invalid=<n>
! energy_MeV_u=<value> AA_fast=<0_or_1> AA_fast_mode=<mode>
! target_A=<A> target_Z=<Z> target_fraction=<fraction>
```

In this file, invalid counts negative or non-finite stored values. The file is diagnostic output only; it is not read back as an AA cache.

8. Operational Interpretation and Troubleshooting

Observed pattern	Likely meaning	Recommended action
Parent calculations close to SR map misses	Parent reuse is failing or Last-Z is repeatedly extended.	Inspect AAcompare compatibility, parent ordering, and requested daughter Z.
Many parent hits, few calculations	Expected extraction of many daughters from each parent array.	No action if results and timing are stable.
Large absent count	Requested daughters fall outside generated AA nodes or Last-Z coverage.	Compare SR filter Z range with G_LastZevaporation and inspect representative absent pairs.
Physical zeros dominate	Nodes exist but carry zero residual cross section.	Compare FAST/non-FAST and NPevap for selected parents.
Unphysical greater than zero	A raw value is negative or non-finite.	Treat as an error; record parent/daughter and stop optimization work.
Equivalence FAIL	Repeated direct request is not stable.	Do not parallelize; inspect global state restoration and AA manager invalidation.
SR map hits stay zero after step 2	Pairs are not being retained within the invocation.	Inspect vmpCS lifetime and parent/daughter indices.

9. Parallelization Assessment

9.1 What is mathematically independent

AA parent calculations for different intermediate projectiles are independent once beam energy, target, Last-Z, requested daughter set, and all AA model settings are fixed. The 252-parent example therefore contains useful coarse-grained parallel work. Daughter extraction within one parent should remain together so one ManagementAA construction serves the full requested daughter list.

9.2 Why direct QThreadPool parallelism is unsafe today

- yieldSecondaryAA temporarily rewrites the global beam and opt->reaction.
- AA FAST handling temporarily rewrites NPevap and G_EvaporationVelocity.
- AbraAblaAA and sAAcompare are shared global cache objects.
- ManagementAA resets GlobalBreak and accesses the global Cible target.
- ManagementAA updates GUI progress, status-bar messages, and processes Qt events.
- Many lower AA routines read additional global physics settings directly.

Conclusion Adding QtConcurrent around the existing parent loop would create races and potentially wrong cross sections. A mutex would make it safe but serial, providing no useful speedup.

9.3 Recommended first parallel architecture: persistent worker processes

Use a small pool of isolated worker processes, each with its own LISE++ global state. The main process builds one job per parent, workers calculate complete sparse daughter maps, and the main process merges results by parent index before the existing multistep propagation. Process isolation avoids global-state races without first rewriting the AA physics library.

Freeze an immutable SR-AA job header: target, energy, Last-Z, FAST mode, AA settings fingerprint, and model version.
Create one work item per active parent, containing parent Z,N and the requested daughter isotope list.
Dispatch dynamically to persistent workers so heavy and light parents balance naturally.
Return raw daughter state (positive, zero, absent, invalid) and scalar cross sections in a sparse result.
Merge results deterministically in parent-index order, then run the existing target-step propagation unchanged.
Keep progress, cancellation, logging, and file output in the main GUI process only.

9.4 Expected performance and limits

If parent construction dominates runtime, four to eight persistent workers should give substantial speedup, but not perfect linear scaling. Initialization, uneven parent complexity, memory bandwidth, and the serial propagation/merge remain. Actual worker count should be capped by memory use and measured rather than set equal to every logical CPU.

10. Recommended Next Improvement

> **Next patch** Add per-parent timing and work-size diagnostics before implementing workers. This is low risk, does not change cross sections, and provides the information needed to choose job granularity and worker count.

For each parent, record:

- Parent Z,N, requested daughter count, and positive/zero/absent counts.
- Whether the parent manager was constructed or reused.
- Elapsed wall time and optional peak/estimated memory.
- Last-Z boundary, FAST mode, NPevap behavior, and settings fingerprint.

The summary should report total AA-parent time, median, 90th percentile, slowest parents, and the fraction of total SR time spent in parent construction. Parallel worker development is justified only if this fraction is large enough to overcome process and serialization overhead.

11. Parallel Acceptance Criteria

- Serial mode remains available as the reference and fallback.
- Serial and parallel sparse cross-section maps match exactly where deterministic, otherwise within a documented numerical tolerance.
- Final target yields and correction factors match the serial baseline across FAST modes and selected NPevap settings.
- Results are independent of worker count and job completion order.
- Cancellation leaves global beam, reaction, Last-Z, and output files in a valid state.
- No worker touches QWidget, status-bar, gauge, or shared output-file objects.
- The 18.7.7 single-step equivalence check continues to PASS in serial and worker execution.

12. Validation Checklist for 18.7.7

- Confirm the first-step plot subtitle identifies calc -> or file + correctly.
- Confirm Secondary AA uses the user's FAST and AA_fastMode settings.
- Confirm the selected Last-Z covers the SR daughter charge range and is restored afterward.
- Check SR map hits/misses separately from AA parent hits/calculations.
- Review positive, physical-zero, absent, and unphysical daughter totals; their sum should explain valid AA requests.
- Require the single-step equivalence diagnostic to PASS before comparing physics results.
- When enabled, inspect SR_AA_intermediate_v1.txt header and representative parent-daughter links.
- For extreme tails, compare production settings against a higher-resolution non-FAST direct AA reference.

Appendix A. Primary Source Anchors

File	Current responsibility
d_Secondary/o_SR.cpp	Run-local vmpCS, yield-path counters, one equivalence probe, optional intermediate writer.
d_Secondary/o_SR_kinematics.cpp	Kinematics map, matching diagnostics, optional intermediate writer.
d_Secondary/d_Secondary_reactions.*	AA-only optional-output checkbox and dialog behavior.
o_Phys/c_EPAX_AA.cpp	Temporary projectile substitution, parent-cache status, direct AA request.
o_Phys/c_EPAX.h	SecondaryAARequestDiagnostics public result structure.
o_Evap/o_AA_manage.cpp	AA parent manager construction and TryGetCS found/zero distinction.
o_Evap/o_AA.h	ManagementAA diagnostic lookup declaration.
d_Evap/d_LastZ_sr.cpp	Secondary Last-Z timeout and silent/fit acceptance behavior.
L_Init/L_Init_options.cpp	User-option persistence for optional intermediate output.
c_Config/c_ConfigPP1.h	LISE project-file persistence for optional intermediate output.
L_Init/lise_version.h	Authoritative 18.7.x release history.

Appendix B. Diagnostic Terminology Quick Reference

Term	Use in 18.7.7
SR map hit	Scalar reuse inside one multistep invocation.
AA parent hit	Compatible ManagementAA parent array reused for a new scalar request.
Parent calculation	Expensive AA parent array required before serving the request.
Physical zero	Requested daughter exists with raw CCSrest equal to zero.
Absent	Requested daughter was not generated in ManagementAA.
Unphysical	Raw cross section is negative or non-finite.
Equivalence PASS	One repeated direct request agrees to relative tolerance 1e-12.