OAK RIDGE National Laboratory

Recent Developments in the R-Matrix Code SAMMY

D. Wiarda, G. Arbanas, J. Brown, M. Pigni, J. McDonnell, C. Chapman, A. Golas*

Oak Ridge National Laboratory

*Rensselaer Polytechnic Institute

NCSP TPR, February 2024

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Summary

- ENDF reading
- Parameter and covariance indexing
- Support for inverse reaction
- Updates to program flow
- SESH updates
- Summary

Switch to AMPX ENDF writing

- Almost all ENDF writing now uses the C++ AMPX ENDF routines.
  - Advantage: One version for all ORNL codes.
    Easier incorporation of new formats
    (we will add external R-Matrix and radius uncertainty shortly)
- Better support (more than one isotope) for LRF=7
- Fixed a previous error with incorrect covariance information written to the ENDF file if pup'ed parameters (fixed but included in data covariance matrix) were used. (Already in use for some new evaluations in ENDF/B-VIII.1-beta)
- To do: allow writing of in the new GNDS format.

Switch to AMPX ENDF reading

- Most ENDF reading now uses the AMPX C++ ENDF routines
  - Exception: Need to add external R-Matrix and radius uncertainty

  To do: Needs to be tested with GNDS, which requires setting a flag. AMPX already supports this.

- Not yet moved: Inputs that only have File 32 (covariance data for resonance parameters) and not File 2 (resonance parameters).
  - In current ENDF, all resonances for which covariances are given are repeated in File 32, GNDS uses links to avoid duplication and ambiguity).
    In cases of a missing File 2, SAMMY assumes the same order of resonances as in the par file.

- The last bullet makes GNDS support difficult – SAMMY input changes are needed. (AMPX C++ ENDF routines can read GNDS resonance information but can't yet save as GNDS).

Parameter and covariance indexing

Assuming W_{12} and E_2 are pup'ed parameters

Parameter	Index	Position in covariance
E_1	1	1
W_{11}	2	2
W_{12}	3	10
W_{13}	4	3
E_2	5	11
....		

The C++ class *CovarianceData* keeps track of the relation between column 2 and 3 (along with the covariance information itself)

The different order in the second column allows for more efficient solves in the adjustment routines.

Previously it was important that the indices in column 1 are strictly in order.
A lot of bookkeeping and checking in SAMMY relies on this fact, a holdover from the time when information between modules was passed scratch files.

As everything is in memory, the order should no longer matter.

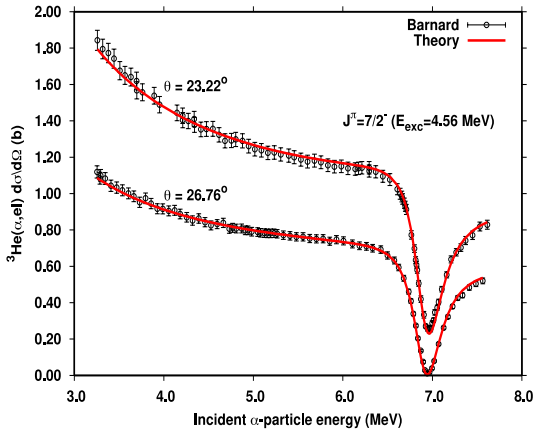
Parameter and covariance indexing continued

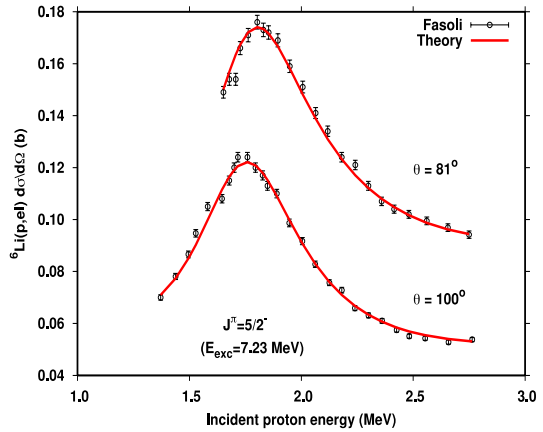
- Why does the indexing matter?
  - Less need to re-organize after input/parameter reading
  - No need to care about order in parts of the code not accessing covariance data
  - Easier to add additional parameters
  - Easier to support new features like inverse reaction support (see next slide).
  - Allows for a better SAMMY covariance format in the future
- Why is this difficult?
  - Lots of internal bookkeeping in SAMMY relies on this defined order.
  - Lots of codes didn't use the index in column 1, but instead counted the parameters anew.
- What was achieved
  - For resonance parameter order does no longer matter.
  - For the remainder we almost finished this task in Q1 of this year and the new code is under review.

Account for inverse reaction

- Often the case of nuclear reactions between light nuclei, the same compound nucleus system can be formed by multiple combinations of projectile plus target nucleus. Two examples are the ^{17}O compound system formed by the interaction of n+^{16}O and a+^{13}C reactions and ^{7}Be formed by a+^{3}He and p+^{6}Li. The inclusion of several reaction systems in R-matrix analyses results in resonance parameters fitted to a set of measured data including multiple incident particle pairs, and, therefore, including a more comprehensive experimental database than a sub-set of measured data for a single incident particle pair.
- Designed to work only with one incident reaction channel, the SAMMY code system was updated to account for multiple incident particle-pair by a simple kinematic conversion. Test cases on the ^{7}Be compound nucleus (see below) were generated.

- Following the update of the SAMMY code, the nuclear data group will be able to expand the nuclear data capabilities to light nuclei evaluations that are of fundamental important for fusion and security applications
- The updated code is available on the public facing SAMMY website.





Updated input handling

- We started to move some of the input handling of the alpha-numeric commands in the SAMMY input into a C++ class.
  - Advantages:
    - Consolidation of interpretation in one place (currently it is read and handled (differently!) up to 3 times). Better checks for duplicate or ambiguous user input.
    - Easier way to add new user options:
      Examples: Inverse channel and new SESH options
    - Ensure that the input is treated consistently
      Example: The same options are now available in angodf as in SAMMY proper
    - Currently the input file is read and re-written in a given SAMMY run. We will eliminate this once all input und parameter options are moved.
- Once finished all user input and parameters are in a C++ class which will allow us to modernize the SAMMY input (we will continue to support the current input) and allow python access to the input options and parameters.
- Only an initial set of commands has been moved. The effort will continue.

Updates to the program flow

Updates to the program flow continued

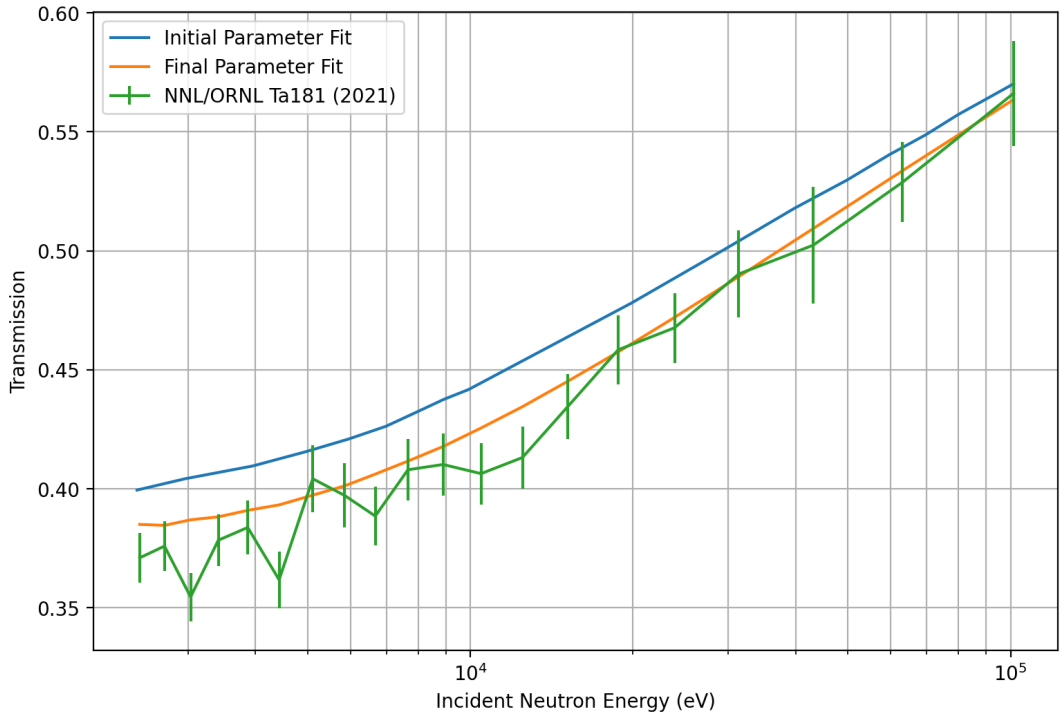
- All resolution broadening functions have a common parent class and the resolution functions used are collected in one module, allowing upstream code to treat these functions as a "black box" that adjusts calculated cross section data and derivatives.
- The same is true for Doppler broadening routines.
- All resonance reconstruction (excepting URR) have a common parent class

The update for the generic resolution broadening functions RSL (a combination of Gaussian plus exponential resolution broadening) to use a common parent class was made difficult:

- The parameters are lumped together with the Doppler broadening parameters in the input or par file. In addition, there are some parameters (recalculated from user supplied parameters) in a different place in the binary covariance file, which are used in instead of the adjusted parameters depending on user input.

SESH+SAMMY updates

- SESH: MC program to estimate res. self-shielding
- Summer '23 intern Alec Golas
  - Incorporated SESH corrections into SAMMY
  - Developed derivative estimation for MC correction factor
  - Implicitly assumes $\frac{\partial C_T}{\partial u} = 0$ for all parameters
- <u>Enables fitting of URR transmission data</u>



$$\langle T \rangle_{measured} = \exp(-n\langle \sigma_t \rangle)\, C_T$$

$$\frac{\partial \langle T \rangle}{\partial u} = -nC_T e^{-n\langle \sigma_t \rangle} \frac{\partial \langle \sigma_t \rangle}{\partial u}$$

Conclusion

- SAMMY is available from https://code.ornl.gov/RNSD/SAMMY Versions will be tagged as needed.
- Support for inverse reaction channels.
- Improved program flow.

This work was supported by the Nuclear Criticality Safety Program, funded and managed by the National Nuclear Security Administration for the Department of Energy.