OAK RIDGE National Laboratory

SAMMY Modernization Efforts

D. Wiarda

A. Holcomb

G. Arbanas

J. Brown

M. Pigni

Oak Ridge National Laboratory

TPR, Feb. 2022

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

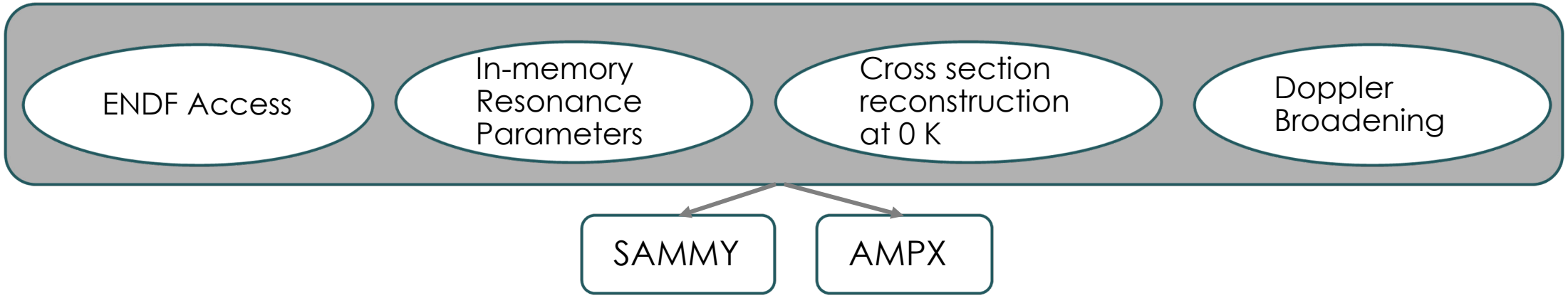
U.S. DEPARTMENT OF ENERGY

Summary

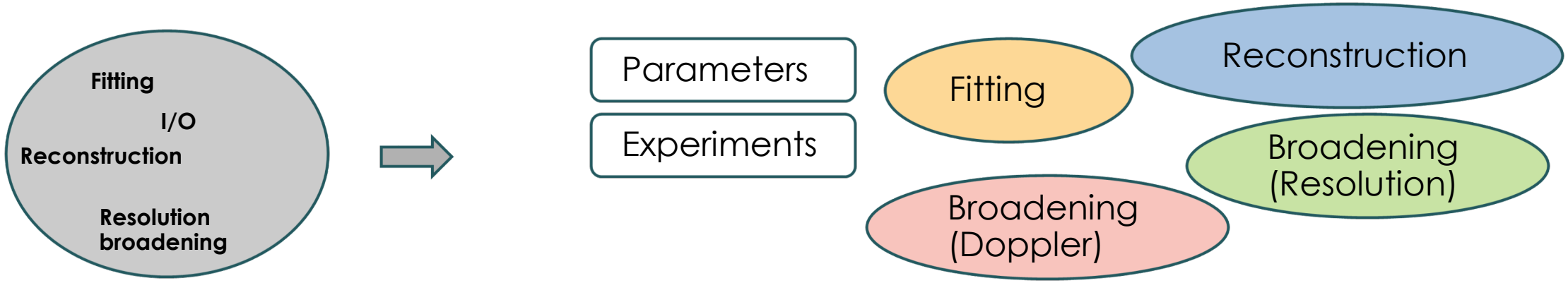
- SAMMY Modernization and Maintenance goals
- Updates to resonance reconstruction
- Updates to derivative handling
- Outlook

SAMMY Modernization and Maintenance goals

- Write code once and reuse



- Transform SAMMY into a modular code, with independent modules with clear interfaces.



- Add new features, which is now easier as only the desired module needs to be changed.

Modularize the 0K reconstruction in the RR

- Currently:
  Sammy has three methods to reconstruct cross section in the RR:
  - Multi-Level-Breit Wigner, old version of Reich-Moore (one isotope, no angular distributions), modern version of Reich-Moore
  - We expect to add the newer AMPX version to this mix, however, it currently does not support analytical derivatives. We expect this to become the default.
- All repeat the same code (loop over energies, transformation to/from transmission, normalization)
- All use global parameters

While Resonance parameters and covariance information certainly are global, they should still be passed in as parameters to a cross section module.

This will give a clean API interface (similar to the one already used in AMPX) to calculate the cross section and derivatives at a given energy and angle at 0K for a given set of resonance parameters.

In the future this will allow us to combine a fit of the RR and the URR.

We created three classes to aid in this project, described in more detail in the following slides.

Parent class for all 0K reconstruction: CrossSectionCalculator:

Initialize:
set the resonance parameters
setEnergyIndependent:
do any energy independent setup
Which_Deriv:
set if and which derivatives are needed
calcCross:
calculate the cross section and derivative for a given energy

An implementations for MLB independent of global parameter has been finished.

The current FORTRAN code was reused were possible.

None of the regression test results were changed

This is the parent class for all supported resonance regions.

This is currently a Fortran class so existing code can be reused, but the interface is designed so that it can easily interact with the API for resonance reconstruction used in AMPX.

Setup 0K reconstruction: CrossSectionCalcDriver

setUpCalculator:
Select the algorithm and instanciate the correct class
calcCross :
pass through to the class set up in setpCalculator
Which_Deriv:
pass through to the class set up in setpCalculator

- All access to 0K is through this class
- Downstream code does not need to know what algorithm is used
- In the future this will allow new algorithms (like access to the AMPX resonance API)

Perform 0K reconstruction: ZeroKCrossCorrections

resonanceRecon:
calculate 0K cross section and derivatives on given energy grid
applyNorm:
apply normalization and convert to transmission if needed
AddParam:
add paramagnetic cross section if desired

Loops over the energy grid and conversions were previously repeated for each algorithm in the RR.

These three classes allow to have an easy access point to calculated the 0K cross section.

As stated, this currently works fully for MLBW. Connection to the other algorithms is in the progress (was completed in Q1 of this fiscal year).

Energy grids in SAMMY

Energy grid C++ class

E_1	Exp. Data 1	Theory 1	$\frac{\partial \sigma}{\partial p_1}$	…
…	…	…	…	…
E_N	Exp. Data 2	Theory 2	$\frac{\partial \sigma}{\partial p_1}$	…

Previous progress report: Developed an in-memory class for energy grid and derivatives and used it in all fitting routines.

- SAMMY uses two energy grids:
  - Experimental data grid
  - Auxiliary grid used for resolution broadening
- In the context of a fit: Need cross section and derivative at each energy and angle for which experimental data exist
- A class (*GridData*) holds the data as outlined above
- A class (*GridDataList*) holds one or more instances of *GridData*

Extension to the energy grid - Motivation

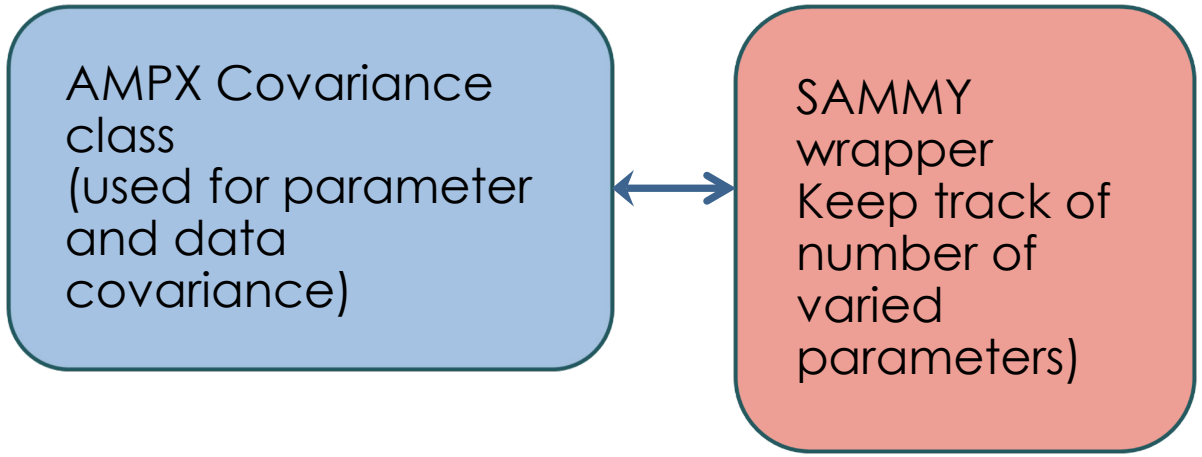
- Experimental samples most likely contain more than one isotope
- Derivatives with respect to the resonance parameters are non-zero only for the isotope to which the resonance parameters pertain (except in the case of fitting inverse reactions)
- Derivatives with respect to normalization or flight path length are non-zero for all isotopes in the sample
- If considering multiple scattering or broadening, cross sections are needed for each isotope during calculation.
- Depending on the number of isotopes involved, this can lead to many derivatives that are zero and need to be carried throughout the calculation.
- In order to allow for a zero-suppression SAMMY treated those two types of derivatives separately. This requires a defined order of the adjusted parameter. In addition, every part of SAMMY needed to be aware of these two types of derivatives.

Extension to the energy grid - Implementation

- Added new class : *DerivativeList*
- The class
  - holds a *GridDataList object, which in turn holds a GridData* object for each isotope.
  - holds a *GridData object to take isotope specific derivatives*
  - *some local parameters needed to keep track of these two objects*
- *This ensures:*
  - *Derivatives that are isotope dependent are only stored for that isotope only*
  - *Calling code does not need to know whether values for the derivative are zero-suppressed or not.*
- *Advantage: Needed complexity is handled once in the new class and the remainder of SAMMY is shielded from it.*

This has been implemented in the MLBW part of the code and needs to be extended to all of SAMMY (As of Q1 of the current fiscal year, this has been done).

Covariance data



Previous progress:

- All covariance information is stored in-memory, data as well as parameter.
- All scratch files related to covariance information, except for the external SAMMY covariance files, have been eliminated.
- Current values of the varied parameters are also held in an in-memory class.

The class is now used throughout SAMMY. It is also used to keep track of the derivatives saved in all the C++ classes used to store derivatives.

If a given derivative is not big enough to contribute to the final fit results (for example narrow resonances far away from the energy range to be fitted) the user has the option to not calculate these derivatives.
Previously, in order to save memory, these derivative were not counted in the number of parameters passed to the fitting routine. This makes the bookkeeping extremely difficult and error prone. Several production defects (now fixed) were linked to this problem.

All parameters are now treated equal, which also makes treatment of linked parameters (all radii for an isotope or linked gamma width) much simpler.

In the future it will make the order of adjusted parameters irrelevant.

SAMMY and GNDS – future plans

- SAMMY has its own ENDF reading and writing routines.
- We plan on switching to the AMPX reading and writing routines. This was delayed in favor to use in-memory C++ classes for resonance parameter and all covariance information.
- Having the relevant information in the in-memory classes will make it much easier to switch out the reading and writing in favor of AMPX methods.
- Switching to these routines will bring GNDS support to SAMMY

Theoretical and experimental data

- Currently SAMMY uses ODF files (an old ORELA style data format) both to read and store experimental and final adjusted data.
  - For user input and output they can be converted to ASCII listing, but internally ODF files are used.
- We would like to not use these any longer – therefore this will be the next enhancement to tackle.
- These are 1-D x-y style data or 2-D double differential style data. **Low level GNDS data containers are ideal to store these data.**
- The GNDS structure on top of these containers allows to indicate the type of cross section.
- Covariance information on these data can already be stored in GNDS (bin-width might have to be the same as the energy values).
- The use of appropriate style element will allow to save the resonance parameters as well as the reconstructed data (Fudge already does something similar) for easy inter-code comparison.

Resonance parameters

- Basic resonance parameters will be read/written into GNDS format. This is needed for the exchange of resonance evaluations with the data community.
- New format extensions will be tested with SAMMY and AMPX. This will allow testing in transport codes.
- In addition:
  We would like to use the GNDS resonance structure for normal SAMMY user input. But we need to determine where flags for adjusted and propagated uncertainty parameters (PUPs) will go.
  - Conversion from and to current input and par file structure will be provided.
  - New features will only be available in the new GNDS style input.

The hope is that we as a community develop tools that will allow to compare and visualize the parameters much easier. This of course includes flags and values pertinent to the type of resonance parameters and calculations.

Resolution broadening and experimental effects

- In order to replace the current SAMMY input with respect to parameter information we additionally need (all of which may have covariance information):
  - Doppler broadening parameters
  - Resolution broadening, for example: parameters for ORELA or RPI beam conditions
  - Sample information.
  - Linking between parameters (for example: all gamma widths of a spin-group should be adjusted together).
  - …
- We haven't decided yet what the best format is for these parameters.

Final outputs from a SAMMY evaluation

The final end-product of an evaluation will continue to be an ENDF file (GNDS or ENDF formatted) containing the resonance parameter and associated covariance.

The use of GNDS style input files for SAMMY, does not mean that these are ENDF libraries to be sent to the data centers, but rather a convenient input layer with re-usable data containers that are used in the data community.

It would be nice to capture as much as possible of the experimental data (possibly a link to a the new EXFOR format (WPEC Subgroup 50), which are used in the evaluation) and some of the information that makes the evaluation easily reproducible, in connection with WPEC Subgroup 49.

SAMMY Release

- SAMMY source code is available from https://code.ornl.gov/RNSD/SAMMY
- The code currently needs SCALE 6.3 and up to compile, instructions are provided.
- Stable versions will be tagged and distributed as before. These official pre-compiled release version will not require a SCALE installation.

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