ORNL/SPR-2019/1360

NCSP Analytical Methods Subtask 3, AMPX Development and Maintenance, and NCSP Nuclear Data Subtask 6, SAMMY Modernization

Dorothea Wiarda
Goran Arbanas
Andrew M. Holcomb
Marco T. Pigni

September 2019

ORNL/SPR-2019/1360

Reactor and Nuclear Systems Division

NCSP ANALYTICAL METHODS SUBTASK 3, AMPX DEVELOPMENT AND MAINTENANCE, AND NCSP NUCLEAR DATA SUBTASK 6, SAMMY MODERNIZATION

Dorothea Wiarda
Goran Arbanas
Andrew M. Holcomb
Marco T. Pigni

September 30, 2019

Prepared for the
National Nuclear Security Administration
Nuclear Criticality Safety Program

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831-6170
managed by
UT-BATTELLE, LLC
for the
DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

LIST OF FIGURES

ACRONYMS

API	application programming interface
CE	continuous energy
CSEWG	Cross Section Evaluation Working Group
ENDF	Evaluated Nuclear Data File
GNDS	General Nuclear Data Structure
GUI	graphical user interface
IAEA	International Atomic Energy Agency
NCSP	Nuclear Criticality Safety Program
NEA	Nuclear Energy Agency
NNDC	National Nuclear Data Center
NRC	US Nuclear Regulatory Commission
ORNL	Oak Ridge National Laboratory
URRR	unresolved resonance region
WPEC	Working Party on International Nuclear Data Evaluation Co-operation

1. INTRODUCTION

SAMMY[1] is a production code that still contains a lot of old-style FORTRAN (F77) code. While the code is widely used and performs well, the legacy code base makes it difficult to maintain and to add new features. However, because it is a production code, it cannot lose any functionality while being modernized. In addition, new features need to be added to support the changing needs of the evaluators to analyze data obtained with new equipment and at higher resolution. In addition, as the data are analyzed with higher fidelity and new models within SAMMY, the processing codes and formats must keep pace. This will ensure that any new SAMMY evaluation can be processed by AMPX and can be tested and used in transport codes like SCALE[2] for maximum benefit of the Nuclear Criticality Safety Program (NCSP) mission.

This effort includes the following overarching goals:

- Reorganize the source code repository to ensure that any source code shared between SAMMY and AMPX[3] is maintained and version controlled in a single location. Functionality shared by SAMMY and AMPX includes cross section reconstruction at 0 K, Doppler broadening, and Evaluated Nuclear Data File (ENDF) input/output (I/O), among others.
- Transform SAMMY into an object-oriented modular code base to divide conceptually distinct functionality into separate modules. For example, the cross section reconstruction, resolution broadening, and resonance parameter optimization presently tightly coupled in SAMMY will be separated into independent and interoperable modules. This modularization will also enable code sharing by separating cross section calculations from the Bayesian optimization functionality presently integrated in SAMMY.
- The modular code base will allow for the addition of new features. Some new features can be incorporated before SAMMY is completely modularized. For example, as soon as the 0 K reconstruction is separated, the complex radius or other additions to the R-Matrix can be added to SAMMY, as it will only affect cross section reconstruction (and derivatives) at 0 K.

SAMMY already contains a suite of test cases that are run continuously to verify that code updates do not change existing behavior or results. However, if any code errors are encountered during the planned modernization, changes to the test cases will be manually inspected and assessed, and test cases will be updated accordingly.

The AMPX code is used to generate all cross section libraries for SCALE. As the ENDF community switches to a new ENDF format to allow for more fine-grained information to be included in the ENDF files, AMPX must keep pace with the new formats. The above-mentioned code sharing will also allow SAMMY to use the new ENDF format. While this will not show an immediate benefit, it will allow for SAMMY to add model parameters like complex radii or correlation of resonance parameter between isotopes into the ENDF files. It will also allow for AMPX to be able to immediately process the new features, making them available to SCALE and the wider nuclear data community. Additionally, if new ENDF formats are proposed to the Cross Section Evaluation Working Group (CSEWG) or the Working Party on International Nuclear Data Evaluation Co-operation (WPEC) community, then an example of the new formats working in a processing code is usually required. The connection between SAMMY and AMPX will ensure that this example code is available once the feature has been implemented and tested by the evaluators.

2. SAMMY REPOSITORY

SAMMY was moved to a new code repository at code-int.ornl.gov. This allows more of the infrastructure in place for SCALE to be used. AMPX is in the same repository as SCALE, allowing for code common to SCALE and AMPX to be shared. SAMMY can benefit from the ample infrastructure established for SCALE, which allows for running continuous integration pipelines. This allows SAMMY to be automatically compiled, with all unit and regression tests executed and reported back on several platforms. The process was started by setting up the pipeline on LINUX using one configuration, and other pipelines are currently being developed.

In the new repository, updated code is not immediately put into the master branch. Instead, it can be reviewed and tested before being merged into the main code branch. This makes it easy for the reviewer to see the proposed changes and communicate suggested changes to the developer. These discussions are archived in the code repository. At least one independent reviewer must review the new code before it is merged into the main branch. This has been an effective approach for ensuring better code, and it has allowed all developers to be aware of changes in the code base. New staff members are now engaged in the review process, allowing them to be exposed to the SAMMY code base. This helps ensure that more staff members are familiar with the SAMMY code and can productively contribute to the modernization effort and development of new features.

The new code repository also includes an issues page for tracking all new features, issues, and bug fixes. All issues and bugs are currently tracked in this repository. Progress on the new features are tracked by linking specific code developments to the feature.

3. CONTAINER ARRAY

Much of SAMMY's code base is still built on the initial implementation. When the code was initially written, FORTRAN did not support dynamically allocated arrays, and memory was limited. This made some design choices necessary to allow for the analysis of large data sets. With faster computers and the availability of more memory, efforts can now focus on making SAMMY easier to read and maintain. The two main components in SAMMY that are attributable to the initial design choice are (1) the *container array*, and (2) the writing of various scratch files. The need for scratch files will diminish as the code base becomes more modularized and parameter storage is moved to C++ in-memory structures. The container array is a direct consequence of the limitation on memory management in FORTRAN 77. The program is compiled with a large array of a fixed size from which pieces are allotted to arrays as needed during calculations. There are two drawbacks to this approach: (1) it requires SAMMY to deal with allocating and releasing memory chunks of the container array when the task could have been passed to the compiler or the operating system, and (2) tools like Valgrind[4] cannot detect if memory outside the array given to the function is accessed. This hampers error detection, which are detailed later in this document. In addition, it limits the size of the problem that SAMMY can analyze unless the code is recompiled with a larger initial array. Due to the way that positions in the array are addressed, there is an upper limit on the array size, even if the computer running the problem has more memory available. In fact, some evaluations have already encountered this problem when extending the upper limit of the resonance range in isotopes such as ^{238}U. Since the code is compiled with a large initial array, a large amount of memory is always needed, even if the actual problem requires less memory.

Due to these issues, the decision was made to use FORTRAN to dynamically allocate the arrays. While this is less efficient than using the allocation from a large existing array, it provides better error detection and more granularity on the memory used. If the extra computational overhead is too large, then a different system can be used for parts of the code if necessary. This approach will be used if performance

tests indicate the need for it, but it is expected that modernization and modularization will diminish this issue. In principle, removing the container array should be an easy substitution. The process requires renaming the array, allocating the array, and changing the name everywhere in the code. However, in practice, this approach leads to many test case failures. The reason is twofold: (1) errors in the original programming lead to overstepped array bounds, and (2) arrays are reused and reallocated in unrelated parts of the code. In several test cases in the current production code, both of these issues have already been found to cause incorrect results. These cases are being investigated to ascertain whether they have an impact on evaluations performed with SAMMY, but the impact is expected it to be small. Any changes in the code that lead to changes in the test cases will be outlined in SAMMY's upcoming release notes.

4. CODE SHARING

Both AMPX and SAMMY must read and write resonance data and covariance information from ENDF formatted files and then calculate cross section, angular distributions, and derivatives at 0 K. Therefore, it is advantageous to share the code, as there is only one version of the code to maintain, and any new features added in SAMMY to better describe nuclear data will immediately be available in the processing code. Currently there is a limited amount of code sharing since AMPX uses SAMRML, a stripped-down version of SAMMY, to calculate derivatives of the cross section with respect to resonance parameters if processing covariance data. Otherwise, AMPX has native code to reconstruct the cross section for different resonance formalism in the resolved and unresolved range. To make code sharing easier, we need to (1) enable easy access to the one maintained version of the code, and (2) restructure the code base to actually use this version. To solve the first problem, the build system of SAMMY was linked to a compiled version of SCALE, which includes AMPX. The shared code will be included in the distributed source code, so external users do not need to have access to SCALE to use SAMMY. Internally to Oak Ridge National Laboratory (ORNL), the shared code is maintained in the SCALE git-lab repository only. The reason for this set-up is that AMPX has already undergone major modernizations and therefore has existing in-memory structures for resonance parameters, along with robust routines to read ENDF formatted data files. In addition, AMPX includes an initial version of C++ code to reconstruct cross section data from R-Matrix parameters. This C++ code will need to be substantially enhanced to be used in SAMMY. The internal repository structure might change; specifically, SAMMY might move into the same repository as SCALE, but in the case of both SAMMY and AMPX, an official release will include all relevant source code.

To share the code between AMPX and SAMMY, data must be easily passed between them. An effective strategy is to develop an in-memory structure for certain data—for example, a multi-group library in SCALE or resonance data in AMPX—in which the data can be accessed through a defined interface. The on-disk format now becomes less important and can be changed without affecting business logic. The current in-memory storage for R-Matrix data in AMPX only exists in C++, and it lacks some information that SAMMY needs, most importantly, the information regarding which parameters are to be varied. In addition, the internal storage order in SAMMY is not identical to that used in AMPX, which is based on the ENDF format. Therefore, FORTRAN bindings were added for the AMPX in-memory structure by using CIX, a tool provided by SCALE to automatically generate these bindings, and by hand-coding some unsupported features. C++ classes were also added to wrap the AMPX in-memory structure, to add the additional information that SAMMY needs, and to provide indices to the AMPX in-memory data. This division was chosen to provide for a clear separation between the parameters for calculations of cross section data (C++ AMPX in-memory structure) and parameters for fitting, (whether a parameter is to be varied) (SAMMY wrapper classes). These wrapper classes may in the future also be used in AMPX to provide a better indexing of the resonance parameters covariance data, which works well now but will become unwieldy as covariance information becomes available for a wider range of parameters or for correlations between resonance parameters of different isotopes. The task of moving all internal storage

of resonance parameters in SAMMY to the new in-memory C++ classes is almost complete. This includes reading the data from the SAMMY input and parameter files. Once this is complete, SAMMY can be switched to use the existing AMPX routines to read the ENDF file. Routines will be used and added as needed in AMPX to write the ENDF files and General Nuclear Data Structure (GNDS) formatted files. The code to write resonance covariance matrices into ENDF in SAMMY is not very transparent and does not allow for saving data for more than one isotope. A combination of existing AMPX code and new code for better indexing of the covariance matrix will improve the readability immensely. While correlations between resonance parameters of different isotopes are not currently allowed in ENDF and cannot be processed in AMPX, the GNDS format allows them. Using the AMPX writing routines will allow SAMMY to save these correlations. Access to the full SAMMY from the AMPX side will allow these correlations to be processed into covariance libraries for testing in SCALE.

5. GNDS AND ENDF

WPEC at the Nuclear Energy Agency (NEA) established Subgroup 38 (SG38) in 2012 to develop GNDS[5] as a more modern format for the ENDF files. The SG38 has formally ended, and the final specification is currently being finalized. Two new subgroups have been formed to follow up on the development of the new format: EG-GNDS, which will govern the new format, and SG-43, which is tasked with developing an application programming interface (API) to the new format. ORNL continues to be involved in both subgroups.

ORNL participated in the WPEC meeting in Paris, attending the sessions for the two sub-groups mentioned above.[6] In the EG-GNDS meeting, the option to release the GNDS format specification as currently available was deferred, as one delegate requested more time to read the documentation. However, since the governance of GNDS is a multi-national endeavor, a procedure was agreed upon to make future enhancements of GNDS easier to approve. The ENDF/B-VIII.0 library is currently available from the National Nuclear Data Center (NNDC) website in GNDS-1.9 format.

The GNDS specification is given as a tree structure in which each node includes attributes describing some properties, and each node also contains one or more children with additional information. For example, the node for resonance information contains an attribute specifying the formalism to be used, and it contains one or more child nodes giving the actual resonance data. In turn, the resonance node would be the child of a deeper tree containing information about the target and the incident particle. Care was taken to separate file structure from physics, and in contrast to the current ENDF format, all units must be given specifically, but they can be inherited from parent nodes. As in the current ENDF format, the GNDS format includes basic containers for 1-dimensional, 2-dimensional, etc. data. However, there are no longer any limitations on the precision of the data given in these containers. The tree structure makes it easier to add new features to the GNDS formatted file without breaking existing processing codes. To help maintain the format, the format specification is now described in JSON formatted files, allowing the attributes and children to be described for a given node along, with a description. The documentation is generated automatically from these JSON formatted files.

The JSON files and the tools to generate the documentation are maintained on the NEA GitLab server (https://git.oecd-nea.org/science/wpec/gnds/formats, which is not accessible to unauthorized users.

To ensure wider adoption, the community needs tools to read and write the new GNDS format with a readily accessible API. SG-43 is tasked with finding a solution to this problem. A two-day working meeting was hosted at ORNL to discuss this problem with the US groups. Work has begun in AMPX and NJOY[7] to address the API and implementation, and FUDGE already supports GNDS, as it uses the

GNDS format internally for transport codes. AMPX can currently process the resonance parameters and covariance information using classes as described in the previous annual report. However, since there are many classes to be yet implemented to fill the AMPX in-memory structures, and since the format is now described by JSON files, an effort was made to determine whether these classes can be generated automatically from the JSON files. Prototypical code was written to read the JSON formatted files and generate the classes that read the GNDS formatted files. This code has not yet been merged into the main code base. This provides a very low-level implementation of an API that tracks the specification exactly. This API is not convenient for direct use by code that would use the nuclear data, but it provides a very good foundation for a higher level API. Therefore, classes were added that fill the AMPX-related in-memory classes for resonance data and covariance matrices from these auto-generated files. This work made it possible to find and fix inconsistencies in the JSON files before the final release of the GNDS documentation. Work continues to add the classes for the angular distribution data, but the effort was slowed to add a fix to the JSON files (1) to ensure that nodes with the same names but different functionality and different parent nodes do indeed generate two distinct classes and (2) to ensure that the parent class addresses the correct child node.

6. CONVERSION TO ALTERNATIVE R-MATRIX PARAMETERS

An existing alternative parameterization of the R-Matrix does not rely on computation of the level shifts or boundary condition constants, so that one arrives at a more "physical" set of resonance parameters.[8] In contrast to the formal parameterization, the Brune representation provides resonance parameters that correspond more closely to their observable values. For example, the resonance energy in the Brune basis occurs at the peak cross section value for the resonance. While the implementation of Brune's alternative R-Matrix remains for future work, an auxiliary conversion code was implemented in SAMMY to convert between the formal and physical (Brune) basis sets.

While the interested reader is referred to the Brune paper on the conversion[8], a few details about the relevant implementation are provided here. While the transformation is performed in the center-of-mass frame, the input ENDF and output ENDF are reported in the laboratory frame, which is in keeping with the usual convention. During the conversion from the formal R-Matrix parameters to the Brune R-Matrix parameters, several nonlinear eigenvalue problems must be solved The method of successive linear approximations was used in this case, which has been found to be easy to implement and robust for these purposes. The nonlinear eigenvalue is considered converged when the iterative update falls to less than one part in ten billion. Even with different starting guesses for the physical Brune resonance energy, under successive linear approximations, there is no guarantee that a unique solution will be found. It is possible to iterate back to a resonance energy that had already been found. The code includes a tolerance that specifies how different two Brune resonance energies must be to be considered different solutions. Two solutions are considered identical if the relative difference between them is less than a tolerance. If the same solution is found in two iterations, then new starting guesses are chosen randomly from a uniform distribution spanning the evaluation energy range. Before a physical solution is considered to be finalized, the reverse transformation is performed to ensure that an acceptable representation has been found. If the physical parameter set cannot be converted back into the formal basis set, then the physical conversion search process is restarted and is continued until an acceptable representation is obtained in both the formal and physical basis sets. The conversion from the physical set back to the formal set is straightforward, only requiring generalized eigenvalue solutions. Consistency between the formal and physical linear algebra conversions is maintained by using SCALE wrapped LAPACK subroutines.

The algorithm for converting formal parameters to physical parameters was tested using a representative ^{19}F evaluation. The physical parameters were visually compared to the cross sections and found to be in excellent agreement; a representative example of the agreement can be found in Figure 1. The routine for

converting physical parameters back into the formal basis set was then tested using the physical ^{19}F parameters. As expected, the original formal ^{19}F parameters were identical to the round-tripped formal parameters to within the precision loss incurred by the intermediate ENDF formatted disk I/O.

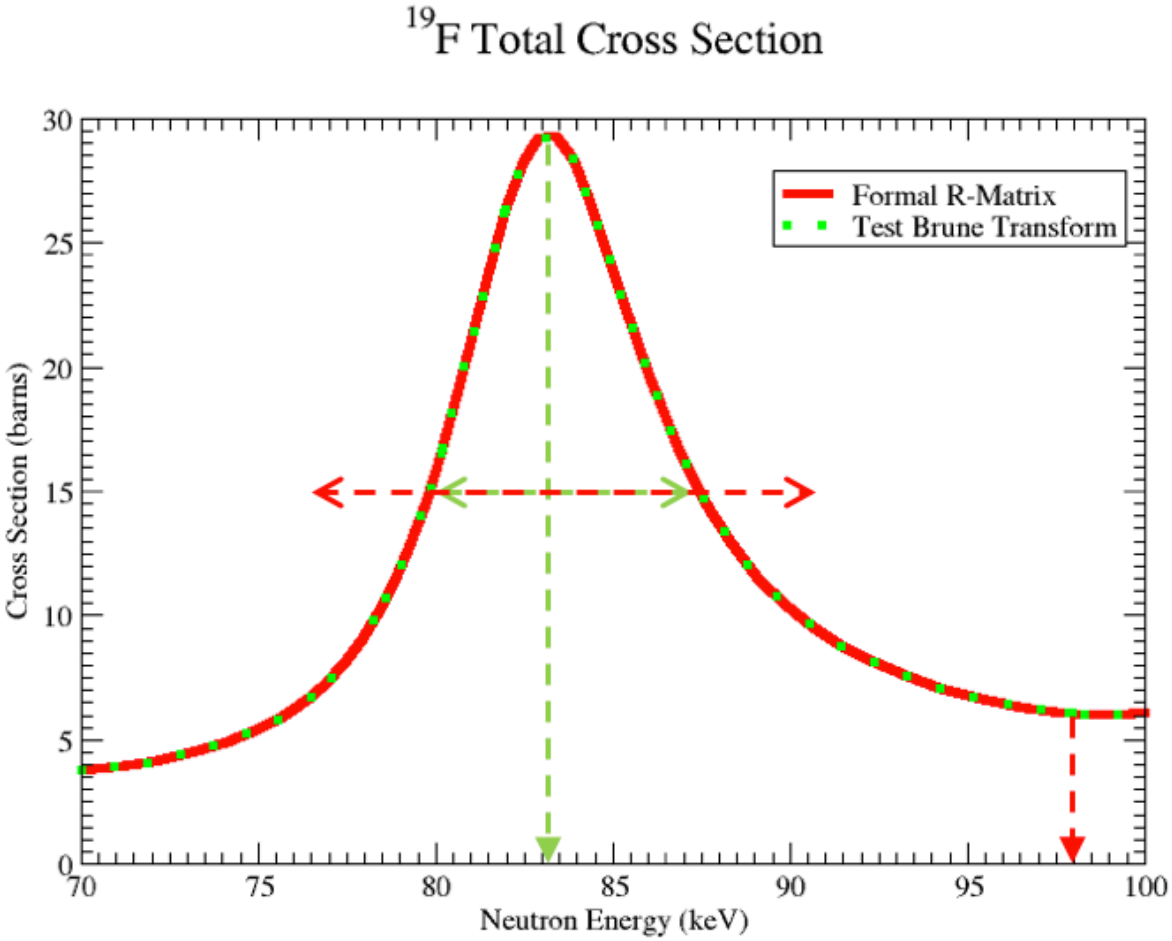

Figure 1. Comparison of the formal and physical resonance parameters for a resonance of ^{19}F.

Red indicates the original formal parameter position and size, and green indicates the Brune representation, which provides a much clearer relation to the observable values in the cross section.

7. LIBRARIES

As part of the upcoming SCALE-6.3 release, new data libraries were prepared, with the initial internal distribution beginning with the SCALE-6.3b3 release, and external distribution beginning with the SCALE 6.3b5 release at the SCALE Users' Group Meeting in August 2019. Data updates that are relevant to the NCSP mission are documented in this section.

With the release of the ENDF-VIII.0 data, the ENDF-VII.0 data are now deemed obsolete and have therefore been removed from SCALE distribution. This includes the removal of the ENDF-7.0 continuous energy libraries, the 238 group multigroup libraries, and both the ENDF-VII.0 200-neutron-47 gamma and 27-neutron-19-gamma multigroup shielding libraries. The old standard composition libraries were

also removed (rev38 and rev39). While the distribution will no longer include these obsolete libraries, the previously released SCALE 6.2 data are still compatible with SCALE 6.3, so users may still run with the older libraries if they have the SCALE 6.2 distribution.

The ENDF-VII.1 data were also modified and augmented as part of this release. Two coupled multigroup libraries were added to support coupled (n,γ) calculations with Polaris. The mg_252n_47g library contains the exact same neutron data that is contained in the neutron-only 252 group library (scale.rev05.xn252v7.1), with the addition of yield and gamma data to support coupled (n,γ) calculations. The mg_252n_47g library uses the usual 47 group gamma structure, which is identical to the gamma structure of scale.rev13.xn200g47v7.1. Similarly, the mg_56n_19g library contains the exact same neutron data that is contained in the neutron-only 56 group library (scale.rev04.xn56v7.1), with the addition of yield and gamma data to support coupled (n,γ) calculations. The mg_56n_19g library uses the usual 19-group gamma structure, which is identical to the gamma structure of scale.rev12.xn28g19v7.1. The ENDF-VII.1 CE libraries were updated with new unresolved resonance region (URR) probability tables to address a normalization issue for a subset of isotopes due to an ambiguity in the ENDF manual. Since the multigroup libraries can use the probability table data to compute Bondarenko factors in the URR, the ENDF-7.1 252 group and coupled 200-neutron-47-gamma were also updated. Lastly, the standard composition library was updated to rev40 to accommodate new data in ENDF-VIII.0.

A new very fine-group library was added consisting of a 1597-group structure based on the AMPX 252-group structure and the MC^2-3 ultra-fine group structures. The energy range from 0.1 keV to 20 MeV was divided into 1,323 groups with an equivalent lethargy width of $^1/_{120}$ to represent broad resonances of intermediate-weight nuclides explicitly. The energy range from 10^{-5} eV to 0.1 keV is represented by 274 groups based on the AMPX 252-group structure. The boundaries of the 1597-group library were chosen so that the fine groups can be collapsed directly onto the 252-group structure. This library was generated for both the ENDF-VII.1 and ENDF-VIII.0 data.

The NNDC published ENDF-VIII.0 on February 2, 2018, and it was subsequently processed by AMPX for use in SCALE applications. The new ENDF data include 556 isotopes (including metastable targets) and 34 thermal moderators (including multiple versions of graphite and hydrogen bound in ice). The ENDF-VIII.0 data were processed to generate continuous energy (CE) libraries for incident neutron and gamma data, and they were deployed in HDF5 format rather than the legacy binary format. The multigroup libraries for incident neutrons were processed analogously to their respective ENDF-VII.1 counterparts. However, the ENDF-VIII.0 versions of the multigroup libraries have additional isotopes and thermal moderators that are not available in ENDF-VII.1. The ENDF-VIII.0 multigroup libraries include a 1597-group library (v8.0-1597), the usual 252-group library (scale.rev01.xn252v8.0), and 56-group covariance data (scale.rev01.56groupcov8.0).

8. AMPX TRAINING

As outlined in the last annual report,[9] AMPX traditionally only ran on LINUX and Mac, mainly due to File I/O. After fixing these issues and providing an initial two-hour training course as outlined in the last annual report, a weeklong training course was presented as part of the larger SCALE training block. The training block included theoretical lectures about nuclear data in general, as well as hands-on exercises to generate multi-group and continuous energy libraries. While the course was not as well attended as could be hoped, it provided a useful initial experience. In conjunction with an NRC project, similar training was also presented at NRC headquarters.

9. FINALIZED BROADEN

In conjunction with an NRC project, a new version of the code BROADEN was finalized for broadening Doppler data. While the modernization was mainly financed by the NRC project, the NCSP portion of the effort included the final testing of the code and the process to establish it as the default version of BROADEN. The code was completely rewritten in C++, and it offers a new API to broaden data, which eventually will allow for switching between different implementations. Currently only Solbrig kernel broadening[10] is supported, but plans are to add support for Leal-Hwang broadening using existing SCALE code. In addition, this broaden API will be integrated into SAMMY to ensure that the same code is used in both packages.

10. INTERCOMPARISON OF R-MATRIX CODES

ORNL staff members Marco Pigni and Goran Arbanas have continued to participate in the multi-year Coordinated Research Project sponsored by the International Atomic Energy Agency (IAEA) to compare the available R-matrix codes.[11] The comparison is currently focused on the R-matrix algorithms, how they are implemented, and what approximations are made. In particular, specific issues such as the impact of relativistic or non-relativistic kinematics and unitarity and boundary conditions are being explored in depth. To this end, R-matrix cross sections for ^{7}Be have been calculated using the same set of well-defined conditions. In this initial successful effort, the calculations were limited to charged particle channels.

11. FUTURE PLANS

The immediate plan for SAMMY is to complete integration of the AMPX in-memory resonance data. This effort will be followed by the switch to using the AMPX ENDF reading and writing routines, which should be finished toward the end of the calendar year. Then the C++ version of the R-Matrix code currently in AMPX will be used to replace the legacy FORTRAN implementation of reconstructing cross section at 0 K. Major changes and updates to this C++ version are expected to be needed to make it fully compatible with SAMMY functional requirements. The initial implementation is expected to be completed in FY 2020. Removal of the legacy FORTRAN container array will be a continuing project throughout the next fiscal year. Once these initial tasks are complete, the remainder of SAMMY will be modularized along similar lines. Data for resolution broadening and doppler broadening will be put into dedicated in-memory structures, and the relevant code will be replaced by its respective C++ implementations. For Doppler broadening, some of the existing AMPX code base will be used. The final goal is to achieve a better separation between cross section reconstruction and Bayes methods[1] to enable implementation of conceptual advances in the Bayes' theorem recently developed by ORNL's Nuclear Data and Criticality Safety Group.[12]

Once the initial tasks are completed and cross section reconstruction at 0 K has moved to C++, then new features for SAMMY can be considered, possibly starting with support for multiple targets or compound systems. The interface used in AMPX already supports the interface functions, but it does not support implementation for this feature. In addition, the impact of complex channel radii will be investigated.

---

[1] This separation of modules would allow for transparent switching between the extant chi^2-minimization method and the Bayesian Monte Carlo resonance parameter evaluation method being developed under NCSP Nuclear Data subtask ND-10.

Sharing the same code in AMPX and SAMMY will allow us to easily propagate the complex channel radii feature into SCALE nuclear data libraries.

For AMPX, work will continue to provide full GNDS support. In addition, work will be done to add processing of double-differential data from resonance parameters for elastic scattering. The data used to be given directly in the ENDF file, but for newer evaluations, resonance parameters are adequate for reconstructing the angular distribution data from the parameters. Code sharing will allow the calculation of the distribution to be leveraged at a given incident energy, exit energy, and Legendre order already available in SAMMY with the reconstruction of a suitable grid in the two dimensions available in AMPX. AMPX uses the graphical user interface (GUI) ExSite, which is based on Java and Netbeans,[13] to generate input files for library generation from templates. It is proposed to change the ENDF reading from Java to the existing AMPX routines and to move the template expansion to C++, thus leveraging templating engines already present in SCALE. In the near future, work will continue to connect these new pieces to ExSite, but the ultimate goal is to combine ExSite with Fulcrum, the GUI used in SCALE, to leverage the existing infrastructure and to provide users with a common entry point to SCALE and AMPX.

12.REFERENCES

1. N. M. Larson, "Updated User's Guide for SAMMY: Multilevel R-Matrix Fits to Neutron Data Using Bayes' Equations," ORNL/TM-9179/R6 (July 2003).
2. B. T. Rearden and M. A. Jessee, Editors, "SCALE Code System," ORNL/TM-2005/39, Version 6.2, Oak Ridge National Laboratory, Oak Ridge, TN (2016)., available from the Radiation Safety Information Computational Center at Oak Ridge National Laboratory as CCC-785 (2011).
3. D. Wiarda, M. L. Williams, C. Celik, and M. E. Dunn, "AMPX-2000: A Cross-Section Processing System for Generating Nuclear Data for Criticality Safety Applications," International Conference on Nuclear Criticality Safety (ICNC 2015).
4. Nicholas Nethercote and Julian Seward, "Valgrind: A Framework for Heavyweight Dynamic Binary Instrumentation." Proceedings of ACM SIGPLAN 2007 Conference on Programming Language Design and Implementation (PLDI 2007), San Diego, California, USA (2007).
5. C. M. Mattoon, B. R. Beck, N. R. Patel, N. C. Summers, and G. W. Hedstrom, "Generalized Nuclear Data: A New Structure (with Supporting Infrastructure) for Handling Nuclear Data," *Nucl. Data Sheets* **113**(12), pp. 3145–3171 (2012).
6. WPEC EG-GNDS Meeting (June 25, 2019), https://www.oecd-nea.org/science/wpec/gnds/meetings/2019_june/documents/agenda.pdf, (accessed October 4, 2019).
7. R. E. MacFarlane and A. C. Kahler, "Methods for Processing ENDF/B-VII with NJOY," *Nuclear Data Sheets* **111**(12), pp. 2739–2890 (2010).
8. C. R. Brune, "Alternative Parameterization of R-Matrix Theory," *Phys.* Rev. C **66**, 04461 (2002).
9. D. Wiarda, A. M. Holcomb, M. L. Williams, and W. J. Marshal, "NCSP Analytical Methods Subtask 3: AMPX Development and Maintenance," ORNL/LTR-2018/1074 (2018).
10. D. E. Cullen, "Program SIGMA1 (Version 77-1): Doppler Broadened Evaluated Cross Sections in the Evaluated Nuclear Data File/Version B (ENDF/B) Format," Lawrence Livermore National Laboratory, UCRL- 50400, Vol. 17, Part B (1977).
11. I. J. Thompson, R. J. DeBoer, P. Dimitriou, S. Kunieda, M. T. Pigni, G. Arbanas, H. Leeb, T. Srdinko, G. Hale, P. Tamagno, and P. Archier, "Verification of R-matrix Calculations for Charged-Particle Reactions in the Resolved Resonance Region for the ^{7}Be System" *Eur. Phys. J. A*. (2019) 55:92, https://doi.org/10.1140/epja/i2019-12753-y
12. G. Arbanas et al., "Bayesian Optimization of Generalized Data," *EPJ Nuclear Sci. Technol.* **4**, 30 (2018), https://doi.org/10.1051/epjn/2018038

13. Netbeans, https://netbeans.org (accessed October 4, 2019).