BNL-112394-2016-IR

Detailed requirements for a next generation nuclear data structure

David Brown

OECD/NEA/WPEC SubGroup 38

July 05, 2016

National Nuclear Data Center
Brookhaven National Laboratory
P.O. Box 5000
Upton, NY 11973-5000
www.nndc.bnl.gov

U.S. Department of Energy
Office of Science, Office of Nuclear Physics

Notice: This manuscript has been authored by employees of Brookhaven Science Associates, LLC under Contract No.DE-SC0012704 with the U.S. Department of Energy. The publisher by accepting the manuscript for publication acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or any third party's use or the results of such use of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Detailed requirements for a next generation nuclear data structure

OECD/NEA/WPEC SubGroup 38*

(Dated: June 28, 2016)

This document attempts to compile the requirements for the top-levels of a hierarchical arrangement of nuclear data such as found in the ENDF format. This set of requirements will be used to guide the development of a new data structure to replace the legacy ENDF format.

CONTENTS

I. Introduction 3
A. Use cases 4
B. How to use these requirements 5
C. Organization of this document 6
D. Connection to ENDF-6 data types 6

II. General Features 8
A. Design philosophy 8
B. Complications 9
1. Is it a material property or a reaction property? 9
2. Different optimal representation in different physical regions 10
3. Ensuring consistency 10
4. Elemental targets 11
5. Legacy data 11
C. Top of the hierarchy 11
D. Particle and/or material properties database 12
E. Documentation 16
F. Representations 20
G. What data are derived from what other data? 21
H. Prototyping functions 22
I. Required low-level containers 25
1. General data/functional containers 26
2. Text 27
3. Hyperlinks 27

III. Evaluations 29
A. `<evaluation>` elements 29
B. "Meta" evaluations 32
C. Reactions and reaction lists 34
1. `<reactions>` and `<summedReactions>` 35
2. `<productionReactions>` lists 38
3. `<reaction>` elements 38
4. Reaction designation 39
D. Cross sections 40
1. Integrated cross sections: $\sigma(E)$ 40
2. Differential cross sections: $d\sigma(E)/d\Omega$ and $d^2\sigma(E)/d\Omega dE'$ 41
E. Products and product lists 41
1. Product list elements 41
2. `<product>` elements 42
F. Distribution and distribution lists 44
1. `<distributions>` and `<distribution>` elements 44
2. Multiplicities 45
G. Resonances 45
1. Designating channels 49
2. Resolved resonances 51
3. Unresolved resonances 52
4. Correcting reconstructed cross sections and distributions 53
5. Expected resonance processing workflow 54
6. How to use the RRR elements in special circumstances 55

IV. Covariance Data 56
A. Covariance definitions 56
B. Covariance between different variables 59
C. Covariance of continuous functions 59
D. Covariance of multi-dimensional functions 59
E. Covariance and correlation matrices 60
F. Weighted sums of covariance 61
G. The "Sandwich Formula" 63
H. Monte Carlo sampling 64
I. Examples of covariance data usage in this hierarchy 64

V. Special Cases 67
A. Atomic scattering 67
1. Incoherent photon scattering 68
2. Coherent photon scattering 68
B. Charged particle elastic scattering 69
1. Legendre series expansion approach: 69
2. "Fake cross section" approach 70
C. Fission reactions 71
D. Fission product yields 73
1. Introduction 73
2. Existing ENDF format 74
3. Detailed FPY requirements 74
4. Discussion of possible implementations 75
E. Spallation reactions 77
F. Radiative capture 78
G. Thermal scattering law 78
1. Introduction 78
2. Theoretical background 79
3. Gaussian approximation of the self part of the scattering kernel 81
4. Coherent elastic scattering 81
5. Incoherent elastic scattering 82
6. Incoherent inelastic scattering in the short collision time approximation 83
7. Discussion of TSL covariance 83

VI. Derived Data For Applications 85
A. General transport data 85
1. Product average kinetic energy and forward momentum 85
2. $\bar{\mu}_{lab}(E)$ 85
3. CDF's from PDF's 85
4. Probability tables in the URR 86
B. Grouped transport data 86
1. Inverse speed 87
2. Multiplicity 87
3. Q-value 87
4. Projectile kinetic energy and momentum 87
5. Bondarenko factors 87
C. Production data 88
D. Radiation heating and damage 89

Acknowledgments 90

* *Edited by D.A. Brown (dbrown@bnl.gov)*

A. Graphical Notation 90

B. Definitions of integrated and differential cross sections 91

C. Terminology 92

D. Contributors 103

References 105

List of Requirements 107

I. INTRODUCTION

It was realized as far back as the Manhattan project that collections of nuclear cross-section data were needed. These collections evolved into the "Barn Book" (Hughes, 1955) first sponsored by the United States Atomic Energy Commission. By 1963, there were many nuclear data libraries such as the United Kingdom Nuclear Data Library (UKNDL) (Parker, 1963) from Ken Parker at the Atomic Weapons Research Establishment, Atomic Energy Authority in Aldermaston, UK; the fast reactor data library from Joe Schmidt at the Institute for Neutron Physics and Reactor Technology, Nuclear Research Center, Karlsruhe, Germany; the NDA library from Herb Goldstein at Nuclear Development Associates, in New York; and the Evaluated Nuclear Data Library (ENDL) from Bob Howerton at the Lawrence Radiation Laboratory in Livermore, California. Each laboratory developed its own storage and retrieval schemes for its data and in many cases libraries were hard-wired into simulation codes. As a result, reactor designers and other data users could not use new cross-section data, even though in some cases the data were available for five or more years (Goldstein, 1968). Furthermore, dissimilarities in the internal formats of each lab kept data users from reconciling differences in calculated values for the same reactor configurations.

There was a need for a common mechanism for intercomparison between these systems. Following a discussion between Henry Honeck (Brookhaven National Laboratory), Al Henry (Westinghouse) and George Joanou (General Atomics) at the Colony Restaurant in Washington, D.C, the Reactor Mathematics and Computation (RMC) Division of the American Nuclear Society (ANS) was requested to sponsor two meetings to discuss a plan to develop this mechanism. Honeck, the chairman of the Division's sub-committee on Evaluated Nuclear Data Files, held these meetings. This effort culminated in a meeting of 18 representatives from 15 US laboratories in New York City on July 19, 1963 to review cross section libraries and discuss means for exchanging these libraries. A sub-committee was appointed to meet in Hanford on September 18-20, 1963 to examine library formats in more detail. The conclusions of these meetings were:

- There was a need for a standard format for evaluated nuclear data.
- The format should be as flexible as possible so that existing libraries could be translated into the standard format and so that the format can be extended to meet future needs.
- This standard format would be the link between a data library and the processing codes.
- It was also suggested that a center be created and tasked with the development and maintenance of the new format called the Evaluated Nuclear Data File (ENDF). This center would also collect and distribute data.

A preliminary version of the ENDF format was sent for review and comment. At the final meeting at Brookhaven on May 4-5, 1964, the 22 attendees discussed changes to ENDF and settled on a final version. The description of the system (referred to as ENDF/A [1]) was documented in the report BNL-8381 (Honeck, 1965). The initial ENDF/A library contained an updated version of the UKNDL library and evaluated data from a number of different laboratories. As ENDF/A did not contain full evaluations, there was also a need for evaluated nuclear data to be used for reactor design calculations. The description of this system (referred to as ENDF/B) grew out of ENDF/A and was documented in the report BNL-50066 (Honeck, 1966). Whereas the format of ENDF/A was flexible, allowing data centers to produce and accept data in a variety of representations, the format of ENDF/B had to be simple and mathematically rigorous to facilitate the development of the supporting infrastructures including data processing, integration and plotting.

Nearly 50 years later, the ENDF format is the "common language" of all major nuclear data projects including CENDL (CENDL, 2009), ENDF (Chadwick, 2011), JEFF (JEFF, 2014), JENDL (Shibata, 2011), and RUSFOND (RUSFOND, 2010). The fundamental need for data has not changed in this time, but the computational tools we have available are much more advanced and our physics understanding has advanced as well. The kinds of data we need to store have also grown markedly. Recognizing the utility of the Working Party on Evaluation Cooperation (WPEC) framework (WPEC, 2012), a WPEC SubGroup (WPEC Subgroup 38, 2012) was formed to coordinate the modernization of the ENDF format in November 2012. Recognizing that processing codes such as NJOY (MacFarlane, 2012), AMPX (Dunn, 2002) and PREPRO (Cullen, 2015) are tied to the legacy ENDF format, this project must including modernizing the support infrastructure. The product of this first WPEC-SG38 meeting were

- The general vision and goals in (WPEC Subgroup 38, 2013a);
- The work plan (WPEC Subgroup 38, 2013b).

In the work plan, the following seven tasks were organized:

[1] ENDF/A referred to both a format and to a library stored in that format.

Requirement 1: WPEC Subgroup 38 tasks

1.1 Low level data containers
1.2 Top level reaction hierarchy
1.3 Particle property hierarchy
1.4 Visualization, manipulation and processing tools
1.5 API
1.6 Testing and quality assurance
1.7 Governance

Furthermore, the vision and goals document (WPEC Subgroup 38, 2013a) lays out the over-arching requirements for the new nuclear data structure:

Requirement 2: Main

2.1 The hierarchy should reflect our understanding of nuclear reactions and decays, clearly and uniquely specifying all such data.
2.2 It should support storing multiple representations of these quantities simultaneously, for example evaluated and derived data.
2.3 It should support both inclusive and exclusive reaction data, that is discrete reaction channels as well as sums over multiple channels.
2.4 It should use general-purpose data containers suitable for reuse across several application spaces.
2.5 It should eliminate redundancy where possible.
2.6 As a corollary to requirements **2**.1 and **2**.2, multiple representations of the same data should be stored as closely together in the hierarchy as feasible.

This document fleshes out the requirements for a hierarchical arrangement of nuclear data discussed in (WPEC Subgroup 38, 2013a) and addresses WPEC Subgroup 38 Tasks **1**.1, **1**.2 and **1**.3. This set of detailed requirements will be used to guide the development of the specifications for a new structure to replace the legacy ENDF format. These detailed requirements do not define the *format* used to store the data. Instead, the structure can be stored in any nested hierarchical meta-language, ensuring that future users and developers of nuclear data will be able to store data in whatever medium is available in the future.

In this document, we will commonly refer to the Generalized Nuclear Data (GND) format and the Fudge code system, two projects initiated at LLNL (Mattoon, 2012; Pruet, 2006). GND is a hierarchical nuclear data format that is a prototype for the system WPEC-SG38 is creating. Fudge is the first code framework that can interpret and manipulate GND formatted data. Neither the GND examples nor the names or arrangements of data in the figures of this document are "set in stone"; all are expected to evolve in the process of developing the new data structure and infrastructure.

The authors of this document have discussed the requirements broadly with members of the nuclear data community and attempted to capture the ideas and needs from a wide range of applications and perspectives. With a high degree of confidence we can say we did not capture all of them. One can surely find one more case of "what about ...". However, we believe this document reflects a comprehensive view of our communities' "best practices" and provides the map to an ENDF modernization.

A. Use cases

A nuclear reaction data library evaluation describes an incident particle (called a projectile) impinging on target. The target may be a single atom or atomic nucleus, or a collection of atoms with which the projectile reacts. The projectile may be the traditional n, p, d, t, ^{3}He, α, γ, or e^- or any other single (composite) particle (e.g., ^{12}C, muon or pion). The ultimate goal for a new structure is to support storing self-consistent, complete evaluations along with their derived data. The derived data include things like deterministic transfer matrices and average outgoing product energy along with the parameters (e.g., group boundaries and fluxes) used in their derivation. Storing derived data in a standardized structure facilitates inter-laboratory comparison of processed data (i.e., derived data used by transport codes). Additionally, the data structure must support uncertainties, generally given as covariance tables, on all these quantities.

The data to be stored are typically a balance between what an evaluator can provide and what a particular application needs. Therefore, it is useful to look at the most common use cases:

- **Particle Transport:** For transport, cross sections for all reactions that are energetically possible over a given range of projectile energy E must be stored, along with a list of outgoing reaction products with the products' multiplicities and probabilities for outgoing energies and angles. Data may be parametric (for example, Watt spectra for storing energy distributions or resonance parameters for storing both cross sections and distributions), or they may be given in tabular form. Some of these data are temperature-dependent and the temperature must be denoted. Also, the rapidly fluctuating cross section in the unresolved resonance region can often be better described as a probability distribution of cross sections. The new data structure therefore needs to support storing cross section probability tables or equivalently multi-band parameters. Additionally, different solution methods (deterministic or Monte Carlo transport) have different derived data requirements:

  - **Deterministic transport:** In addition to cross sections, deterministic transport codes require transfer matrices (which store the double-differential cross section for each reaction product as a Legendre expansion where each term is averaged over incident and outgoing energy ranges), as well as energy and momentum deposition cross sections. Although these quantities are derived from the cross section, multiplicity and distribution data, the transfer matrices in particular are computationally intensive to calculate, so the new structure needs to be capable of storing them for re-use and exchange.
  - **Monte Carlo transport:** The new data structure may need to support storing other cumulative distribution functions (CDFs) for sampling outgoing particles.

- **Transmutation/Isotope Burn-Up:** For isotopic production and depletion, cross sections and optionally outgoing particle spectra for chosen reactions that are energetically possible over a given range of projectile energy E are needed. Fission product yields are definitely part of these needed data. In addition, the decay half-lives and product branching ratios must be known for all nuclei to enable the time dependent calculation of the isotope inventory.

- **Astrophysical network calculation:** In an astrophysical network, one needs isotopic accretion and depletion cross sections. These data may be averaged over Maxwellian neutron spectra to simulate the neutron flux in an astrophysical environment. Astrophysical networks also involve reactions with charged particle projectiles – not just neutrons – and may required detailed knowledge of charged-particle interactions in plasmas.

- **Web Retrieval:** For archival, there are no completeness requirement as the data will not be used in applications. The data will most likely only be visualized. That said, visualization requires a pointwise representation for many data types.

- **Uncertainty Quantification (UQ):** UQ applications encompass all of the above use cases. The defining difference is the need to also specify uncertainties/covariance on aspects of data. One can then either generate statistical realizations of these data (if one is adopting a Monte Carlo approach to UQ) or one can use the "Sandwich Formula" (if one is using a deterministic approach).

For all use cases, data will need to be documented. Therefore we will need facilities for documenting data clearly, concisely and as machine and human readable as possible. In addition, version information for the data structure, data format, the documentation, the evaluation itself, the codes used in evaluation, etc. all need to be stored.

B. How to use these requirements

This document is a list of the requirements for the new data structure. It is not the specifications of the format. At times the details of the requirements may constrain the actual specifications so much that the requirements may seem to be specifications. Other times the requirements will be broad and may be left to interpretation or may have several possible implementations.

In this document, requirements are called out and uniquely numbered so that they can be clearly referenced in later work. A requirements list is formatted as follows:

Requirement 3: An example

3.1 Don't be evil
3.2 Respect the user
3.3 Let the evaluators express themselves clearly

In the process of developing these requirements, we have had discussions[2] among ourselves and with other members of the nuclear science community. We have captured many of these discussions in this text so that future users of the data and data structures can understand our reasoning. Much of this discussion has been already integrated into the requirements. However, there were other questions that required further discussion and these are formatted as follows:

Discussion point:
Users did not get this point.

Resolution:
We added an example to clarify it.

Also, during the effort to create this and its companion documents, it became clear to us (the authors and other contributors) that we often do not agree on the meaning of a concept. Given that the ENDF format manual (Trkov, 2009) is several hundred pages long and encapsulates decades of accumulated knowledge, it is no surprise that we are all not intimately familiar with all of its intricacies. Therefore, we provide a glossary in appendix C.

[2] And they were sometimes contentious

We also tend to use XML in examples and to denote elements in the hierarchy. This is only a matter of convenience as the data should be serializable in any nested hierarchical meta-language (e.g., HDF5, ROOT, JSON, Python classes). We denote major nodes/elements in the XML element-like notation (e.g., `<element>`) and attributes of these nodes/elements without the brackets (e.g., `attribute`).

Many of the data structures and data hierarchies described in this work are illustrated in Crow's Foot notation for Entity Relationships (Barker, 1990). Appendix A summarizes our usage of this notation.

Finally, we note that as we list and discuss requirements, we encounter suggestions for test cases for either data or supporting codes. This are marked up as follows:

SUGGESTED TEST: This is an example test.

C. Organization of this document

This document is divided into six main sections. This, the first, section (§I) is merely an introduction. The second section (§II) lays out the design philosophy of the new nuclear data structure and outlines general requirements as well as the most commonly used elements of the data structure. The third section (§III) delves into the main hierarchy of the data structure, namely those things needed to describe particle transport and reaction data. The fourth section (§IV) details covariance data. The fifth section (§V) deals with many special cases found in ENDF formatted libraries. These special cases fit in the hierarchy detailed in §III, but may have additional physics motivated elements. The last section (§VI) provides a partial listing of requirements for derived data needed for applications.

The ENDF Format manual defined several low level data structures (e.g. TAB1, TAB2, LIST, etc.). This document lists some of the requirements for the structures in §II.I. A more complete listing is available in Ref. (WPEC Subgroup 38, 2015b). We note that there is some difference between the element names used in this document and those in Ref. (WPEC Subgroup 38, 2015b). We hope that the naming in this document is clear enough so that the reader can determine the mapping of names in the two documents.

Many of the subjects dealt with in this text have no direct analog in the legacy ENDF format. In particular,

- *The ability to denote that data are derived from other data* – This is a key part of the new data structure, responding to requirement **2**.2. This feature is discussed in §II.G.
- *Default evaluation and derived representations* – This feature was added to streamline the appearance of data files by allowing the evaluator/developer to state default parameters, e.g., group structures. This is detailed in §II.F.
- *Derived data* – Each institution performs post-processing of evaluated data differently. By storing these derived data in the nuclear data structure, we enable institutions to share results using a common language. These derived data structures are discussed in §VI.
- *Hyperlinks* – This is a ubiquitous feature of the new data structure and is detailed in §II.I.3.
- *"Meta" evaluations* – We often find the need to connect evaluations from different physical regions together or to assemble new reusable materials in input decks for application codes. The data structure that enables this is described in §III.B.
- *Prototyping new formats/features* – We foresee the possible need to try out new interpolation schemes or functional forms in various data. The structure described in §II.H enables this.

D. Connection to ENDF-6 data types

Those familiar with the legacy ENDF-6 format will find a lot of commonality with data in this new data structure. Here we list the Files (denoted by MF) in the ENDF-6 manual (Trkov, 2009) and explain where one can find the corresponding requirements in this document. In this section we use E to denote a projectile's energy, $\mu = \cos(\theta)$ and E' to denote a product's outgoing angle and energy, respectively, and P to denote a probability.

MF=1: *General information*

- **MT=451:** *Descriptive data* – Unlike the legacy ENDF format, documentation is allowed at nearly all levels in the new data structure. See §II.E for a discussion of the `<documentation>` element.
- **MT=452,455,456:** *Number of neutrons per fission* – These are product data of a fission reaction. §V.C discusses fission reactions, §III.E.1 discusses product elements and §III.F.2 discusses particle multiplicities.
- **MT=458:** *Energy release due to fission* – This is product data of a fission reaction. §V.C discusses fission reactions, §III.E.1 discusses product elements.
- **MT=460:** *Delayed photon data* – This is product data of a fission reaction. §V.C discusses fission reactions, §III.E.1 discusses product elements.

MF=2: *Resonance parameters* – See §III.G for the discussion of the `<resonances>` element.

MF=3: *Reaction cross sections*

- *Neutral particle scattering* – All cross section data structures are described in §III.D.
- *Background cross sections in the resonance region* – All are stored with the resonances themselves as described in §III.G.
- *Charged particle scattering* – These cross sections are treated in the same manner as neutral particle scattering for the most part. Only the elastic `<reaction>` is tricky because of the divergences in the differential cross section. This special case is explained in §V.B.

MF=4: *Angular distributions* – These are product data of a reaction. The enclosing elements are described in §III.F.1.

MF=5: *Energy distributions* – These are product data of a reaction. The enclosing elements are described in §III.F.1.

MF=6: *Product energy-angle distributions* – These are product data of a reaction. The enclosing elements are described in §III.F.1.

MF=7: *Thermal neutron scattering law data* – This data describes scattering off of macroscopic materials by very low energy neutrons. This data has many unique requirements that are covered in §V.G.

MF=8: *Decay and fission product yields*

- *Decay data* – These are particle properties, see §II.D and reference (WPEC Subgroup 38, 2015a).
- *Fission product yields* – Although this is product data of the fission reaction, it has unique and detailed requirements covered in §V.D.

MF=9: *Multiplicities of radionuclide products* – This is product data of a reaction. §III.E.1 discusses product elements and §III.F.2 discusses particle multiplicities.

MF=10: *Production cross sections for radionuclides* – Production cross sections are described in §VI.C.

MF=12: *Photon production yield data* – This is product data of a reaction and §III.F.2 discusses particle multiplicities. However, the branching fractions from discrete level de-excitation are particle properties and covered in §II.D and in (WPEC Subgroup 38, 2015a).

MF=13: *Photon production cross sections* – Production cross sections are a form of derived data and are described in §VI.C.

MF=14: *Photon angular distributions* – This is product data of a reaction. The enclosing elements are described in §III.F.1.

MF=15: *Continuous photon energy spectra* – This is product data of a reaction. The enclosing elements are described in §III.F.1.

MF=26: *Secondary distributions for photo- and electro-atomic data* – This is product data of a reaction. The enclosing elements are described in §III.F.1. In §V.A, we list additional issues encountered while storing atomic reaction data.

MF=27: *Atomic form factors of scattering functions* – Atomic data requirements are discussed in §V.A. Scattering functions and form factors modify cross sections for photon scattering and the cross section elements are described in §III.D.

MF=28: *Atomic relaxation data* – These are particle properties, so see §II.D and reference (WPEC Subgroup 38, 2015a).

MF=30: *Covariances of model parameters* – See §IV for a description of the unified covariance structure, in particular the discussion of the "Sandwich Formula" in §IV.G.

MF=31: *Covariances of fission* $\overline{\nu}$ – See §IV for a description of the unified covariance structure.

MF=32: *Covariances of resonance parameters* – See §IV for a description of the unified covariance structure, in particular the discussion of the "Sandwich Formula" in §IV.G.

MF=33: *Covariances of neutron cross sections* – See §IV for a description of the unified covariance structure.

MF=34: *Covariances for angular distributions* – See §IV for a description of the unified covariance structure.

MF=35: *Covariances for energy distributions* – See §IV for a description of the unified covariance structure.

MF=40: *Covariances for radionuclide production* – See §IV for a description of the unified covariance structure.

II. GENERAL FEATURES

In the sections to follow, we describe the philosophy behind the data structure we are developing and some of the complications of this endeavor. We then layout the top nodes of the hierarchical structures and describe several repeated patterns in the proposed data hierarchy. These repeated patterns or motifs include documentation elements (see subsection II.E), lists of reaction products and the sub-elements of a `<product>` specification (see subsections III.E.1, III.E.1, III.F.1, III.F.2) and more complicated constructs detailing what data are derived from what other data (see subsection II.G). We note two other reoccurring "patterns": covariance data (described in section IV) and Fission Product Yields (FPY). FPY data exists for both induced reactions (for example, the Neutron Fission Product Yield sub-library in ENDF/B-VII) and as decay products from spontaneous fission (in both the ENDF/B-VII Decay and Spontaneous Fission Product Yield libraries). FPY data structure requirements are detailed in section V.D. We comment that the motifs described here are also needed for the particle properties data.

A. Design philosophy

The requirements listed in Requirements **2** give the overarching goals that we believe a new data structure should achieve. These goals represent the most broadly held beliefs that guide all subsequent requirements. Some of these goals may seem contradictory. For example, allowing multiple representations while at the same time eliminating redundancy appear to be conflicting goals. However, derived data are not redundant: they reflect the choices of the processor and needs of particular applications (e.g., group boundaries and fluxes) in addition to the original evaluated data.

It is up to each evaluated data project to determine specific requirements, such as the completeness, of the data to be stored in a particular library. Similarly, it is up to the data processor to decide how to process the data, but the resulting processed data need to be stored in a common structure to facilitate exchange and comparison. Crafting a structure that is capable of balancing all of the goals in Requirements **2** is the task at hand.

As implied in the first requirement for the new data structure, data shall be stored in a hierarchy. The design choices made in the development of the original ENDF format enabled the community to shoe horn data onto punch cards at the expense of obfuscating even the simplest things such as determining when one floating point number ends and another begins or dropping the data units. The inability to read a data file without a deep understanding of the ENDF format leads to issues with quality assurance. Nevertheless, there is a common sense arrangement of data that reflects both our understanding of particle transport and reaction physics. Capturing and encoding that arrangement within the formal structure will vastly improve the usability of the data for those with little knowledge of our data structures. This is an obvious benefit to the community at large, enabling more people to work more effectively and with greater assurance that they are correctly using the correct and best data for their applications.

Hierarchical storage defines itself by means of nodes existing as parents, siblings and children spanning logical levels. We require a means of identifying these nodes and do so by providing them names. These names are, to some degree, arbitrary and thus likely to be contentious. Care should be taken to ensure that names align themselves with generally accepted definitions. At the higher levels, these names should convey the physical essence of the data, for example `crossSection`. Additional qualifiers and context can help (e.g. a `<distribution>` within a `<product>` is clearly a particular product's distribution). At the lower levels, they should convey the type of data being stored, for example points defining a piecewise continuous curve or vectors defining bands in a sparse array.

There are two significant questions regarding the naming of nodes: the choice of language and the character set used to encode them. These are pragmatic choices. The character sets available across storage systems vary greatly. Choosing a restricted set will enable a broader adoption of this structure into working implementations. As many of these storage systems allow only ASCII characters, or only a limited set thereof, the allowed language and character set need to be clearly defined.

The data structure must store data and information qualifying the data. The qualifying information are typically known as meta-data. There are values that are clearly data; for example cross sections, resonance parameters or emission distributions. There are also values that are clearly meta-data; for example the number of point-wise energy/cross section pairs, the units for values or the frame of reference for a particle emission. Hierarchical storage systems typically allow attribute names/values to be associated with a node. In general, the data structure should store meta-data as attributes to nodes and data within the node itself.

At the bottom of such structure are the actual data. These are general numeric—real, integer, complex—values with documentation, annotation or meta-data provided as text. At some point these values must be laid out sequentially in a storage system; that is, they must be written into a file for archival storage or exchange. At this lowest level, it is vital that the beginnings and ends of each component of the data be clearly delineated and that the necessary mathematical transformations and inherent physical meaning of the data be defined by the structure containing these data. Anyone who has ever struggled to

determine where the data they need are stored inside a large data file will understand just how important this requirement is.

Requirements related to the hierarchical structure are:

Requirement 4: Hierarchical structures

- **4**.1 Data shall be stored in a hierarchical structure
- **4**.2 At least one implementation of the data structure shall be made using text, not binary, storage
- **4**.3 Node names should be used to convey meaning
  - **4**.3.1 High-level nodes should convey physical meaning
  - **4**.3.2 Low-level nodes should convey mathematical or functional meaning
  - **4**.3.3 Node names shall use a restricted character set to work across varying storage systems
- **4**.4 Attributes should be used to store meta-data
  - **4**.4.1 Attribute names shall use the same restricted character set as node names

B. Complications

As we develop requirements for a new nuclear data hierarchy, we naturally encounter thorny issues that must be dealt with before specifications can be written.

Differences in opinions and advancement in understanding can lead to a desire to restructure existing data layout. As we consider solutions to these issues, we must strike a balance between keeping the legacy solution – that is, the ENDF-102 (Trkov, 2009) with which the nuclear data community is already familiar – and the desire to reach for new solutions. We also comment that we will need to strike a balance in how deeply to nest the hierarchy since some storage schemes perform better with a flatter hierarchy (e.g., HDF5) even though a deep hierarchy may make sense for organizing the data more clearly.

1. Is it a material property or a reaction property?

Some kinds of data can be viewed as reaction-independent properties of a particle (e.g., projectile, product, target material). A case in point is the gamma branchings from an excited nuclear state of a nucleus. An excited nucleus decays via a gamma cascade through the lower levels of the nucleus in accordance with the tabulated gamma branchings, independent of whether it was formed in a neutron induced reaction or fission or any other process that leads to the same compound system.

Given this, we view *all "particle/material properties" as data that are independent of the excitation mechanism.* This includes (but is not limited to):

- For atomic nuclei:
  - target mass
  - number of neutrons, protons (and maybe even hyperons!)
  - nuclear level schemes (energies, spins, parities, ...) below the neutron separation energy[3]
  - level lifetimes, level widths
  - gamma and decay branching ratios from particles emitted during the de-excitation of excited states of a nuclear level
  - emission spectra from these nuclear decays
- For elements:
  - Z
  - Chemical symbol and name
- For atoms/ions:
  - mass
  - atomic shell properties (binding energies, spins, parities, ...)
  - X-ray decay and branching ratios from excited states
  - level lifetimes and widths
  - emission e^- spectra from the internal conversion of gamma rays emitted from nuclei
  - charge state
- For composite materials (as encountered in thermal neutron scattering):
  - target density (at Standard Temperature and Pressure)
  - target stoichiometry

These lists may be amended as needed in the discussion below and a deeper discussion of them will be presented in the requirements for the material properties database. Indeed, it was recognized at the December 2013 WPEC meeting that, in order to ensure consistency of masses, Q values, levels and gammas within an evaluation, an external database is needed to perform this role library-wide. This database addresses main Requirement **1**.3 and is covered in section II.D.

A separate particle properties database also provides a mechanism to store the ENDF/B-VII.0 and ENDF/B-VII.1 Decay and Atomic Relaxation sub-libraries (Chadwick, 2006, 2011).

[3] The ENSDF project (ENSDF, 2015) has begun storing information for unbound resonances above the neutron separation energy and these too may someday be included in our particle property descriptions.

2. Different optimal representation in different physical regions

There are different optimal representations of data in different physical regions. For example, at low energies (i.e., in the resolved resonance region), neutron scattering is best described with an R matrix approach, whereas at higher energies where resonances overlap enough to allow Hauser-Feshbach treatment, scattering is better described via tabulated data as shown in Figure 1. This implies, for example, that

- Different physical regions may require us to change our concept of what a target is (e.g., fast neutrons see a single nucleus while thermal neutrons may (in)coherently scatter off many atoms in a material)
- Different macroscopic environments require us to change our description of microscopic data (e.g., Doppler broadening due to temperature effects)
- Different incident energies affect what particles are produced (e.g., pre-equilibrium, multifragmentation, particle production, spallation)

This fact was already recognized in the design of the legacy ENDF format and is a reality we too must confront (Trkov, 2009). Hence, we not only must consider different optimal representations (e.g., resonance parameters) in different physical regions, but we also must consider

- Different alternate representations (e.g., resonance parameters versus reconstructed pointwise cross sections)
- The matching (and potentially overlap) between representations
- A mechanism to "glue" them together, especially in cases where the concept of a target or reaction changes dramatically (e.g., thermal neutron scattering on molecules transitioning to high energy neutron resolving the nuclei in the atoms of the molecule)

3. Ensuring consistency

As we design the data structure and supporting infrastructure, it is important to maintain internal consistency of the data. Within an evaluation, we must ensure at the very least consistency between:

- Cross section sum rules
  - Summing to the total cross section
  - All (n, n') cross sections sum to total inelastic (ditto for other similar reaction types); similar to MT=3
- The sum of the prompt nubar and all delayed nubars must equal the total nubar in a fission reaction
- Masses, Q values, thresholds, upper energy bounds on secondary distributions
- Normalization conditions (on probability distributions, multiplicity tables, etc.)
- Energy and momentum balance
- Energy ranges of tables within a reaction and between different physical regimes
- Gamma branchings (that is, an excited state should have all allowed gamma-decay paths open independently of how it was produced, and their probabilities should sum to 1.0)
- Resolved and unresolved resonance regions and the fast energy region
- Original and processed data

Between evaluations, we must also ensure consistency between

- Fission product yields and decay data linkage
- Fast reaction region and the particle production region if they are stored in separate evaluations as is the case in for example JENDL and the JENDL-HE high energy library.
- Masses and other material properties
- Covariances and mean values between data common to both the Neutron Standards (Carlson, 2009).

Some of these can be handled with a simple hyperlink. Others may require capability within external processing/manipulation infrastructure. In the following discussions, we will point to features of the hierarchy that enable maintaining consistency.

Material properties are a special case where inconsistencies may need to be tolerated. In particular, when dealing with legacy ENDF evaluations, there is no guarantee that the same masses or level schemes are used consistently throughout an evaluation. As such, we may need to override any external material properties database with local versions within an evaluation.

Discussion point:

There is another consistency problem hinted at above: many measurements and evaluations are correlated with the Neutron Cross Section Standards (Carlson, 2009) or other common reaction monitors. Measurements of cross section ratios to a reaction monitor are the most common form of correlation.

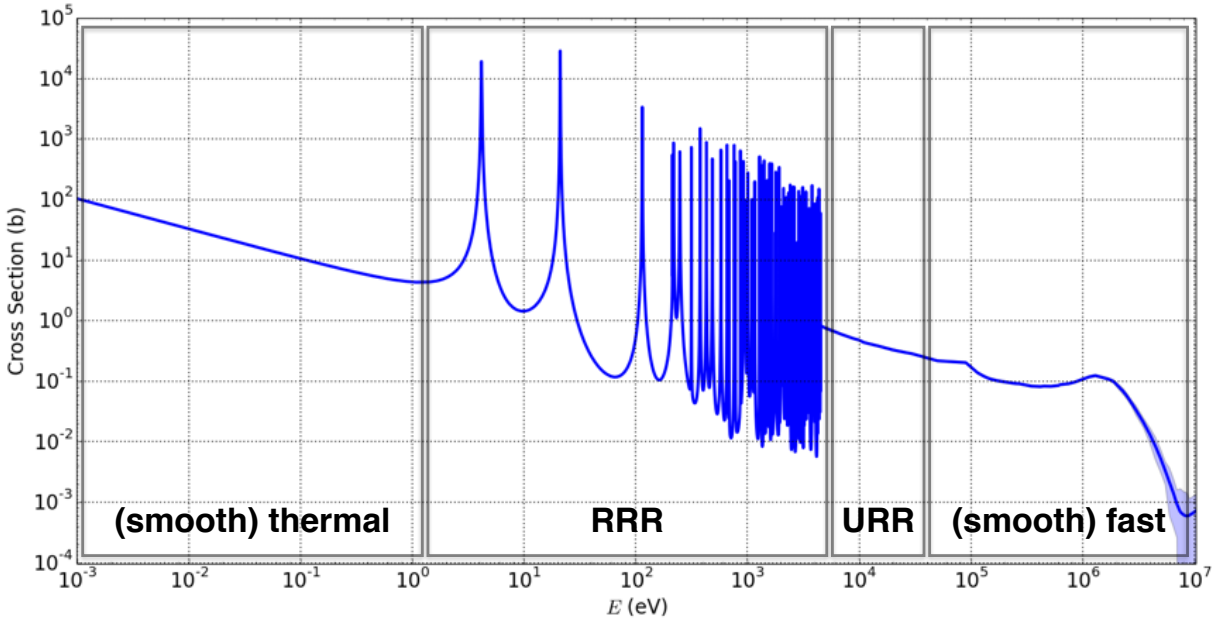


FIG. 1 Cartoon of energy regions in a neutron induced reaction. The high energy region labeled "(smooth) fast" is usually handled in a very different way than the low energy regions where the R matrix approach applies.

In all rigor, one would want a way to update all evaluated files when a standard is updated.

4. Elemental targets

The Neutron Cross Section Standards (Carlson, 2009) and the ENDF/B-VII.1 Data Library (Chadwick, 2011) both contain "natural carbon" (i.e., $^{\text{nat}}$C) as a target. Older libraries have many more natural targets. A natural target is complicated because it represents a weighted average of several isotopes. Questions include:

- How do we know the outgoing isotopics since we cannot deconvolve a reaction's cross section into its isotopic components?
- How can we properly do the kinematics of a collision if the target mass is not well specified?
- What do we use to simulate systems with enriched samples if all we have are elemental data?

Although we cannot remove elemental evaluations from legacy data libraries, we can create a system that deprecates them in future libraries and we can provide a means for constructing elemental targets from individual isotopic files. This scheme would obviously need a provision for containing all information extracted from the accurate measurements on natural targets. Such a system is detailed in §III.B.

5. Legacy data

More than six decades of work have gone into the creation of the nuclear data libraries and formats. Many very old data files are still in production and are needed for specific applications. These data must be supported until such time as they can also be updated.

The following requirements reflect the need to support legacy data:

Requirement 5: Legacy data

- **5**.1 Grandfather in all valid ENDF data in currently supported ENDF-102 Version 6 formats.
- **5**.2 Correct, wherever possible, ENDF format mistakes and inconsistencies.
- **5**.3 Have a process for deprecating data or formats that must be supported but that we intend to phase out at some later date.

C. Top of the hierarchy

In the XML format, only one top level node is allowed per file. As we intend to be able to serialize the new data structure into XML, it makes sense to adhere to this restriction. Currently only six kinds of top level nodes are needed for ENDF style reaction data, but more could be added to support publication-centric arrangements of data such as found in the EXFOR (EXFOR, 2015) or

ENSDF (ENSDF, 2015) data libraries. These six top level nodes are:

`<library/>`: A container type that collects all of the parts that define a nuclear data library such as the JEFF neutron sub-library (JEFF, 2014). This container is discussed below.

`<covariances/>`: A container type that holds a collection of covariance data. This element can be used as a top node when one stores cross material covariance data that is associated with more than one evaluation. See §IV for more detail. In GND-1.7, this element is called a `<covarianceSuite>`.

`<particles/>`: A container type that holds a collection of particles and their properties (see §II.D). This element can be used as a top node when distributing a particle database not already associated with an evaluation or a library.

`<evaluations/>`: A container type that holds a collection of `<evaluation>`s or `<metaEvaluation>`s. It can be used as a top node when distributing lists of `<evaluation>`s or `<metaEvaluation>`s and it is detailed below.

- `<evaluation/>`: This is the collection of data containing everything needed for particle transport and is analogous to an ENDF-6 Material. It is described in §III.A. In GND-1.7, this is referred to as an `<evaluationSuite>`.
- `<metaEvaluation/>`: This is a special container that allows the user to connect other evaluations together in much the same was as can be done with NJOY's `xsdir` file (MacFarlane, 2012). See §III.B for details.

We note that top level nodes are distinguished from lower level elements not just because they can be the root node of a document but also because they also contain default information meant to apply to all nodes within them. We will later term this default information "representations".

There are many cases where one simply wants to collect together evaluations and the like:

1. When defining a (sub)library
2. To provide an index of a set of files including associated particle definitions and covariance data
3. To collect web retrievals under one element
4. To produce a valid XML file containing more than one evaluation (the equivalent of an ENDF TAPE).
5. To modify a previously-defined ¡library¿, e.g. by replacing one or more evaluations with new test versions

To enable the first two of these uses, we must provide the top-most level element `<library/>`. This allows a library project such as JENDL (Shibata, 2011) to group together everything needed to define an evaluated data library. This element is illustrated in Fig. 2. Within a library, there exists a list of evaluations contained in the `<evaluations/>` element. By separating out this list, we can meet needs #3-4. This element is illustrated in Fig. 3. The requirements for `<library>` and `<evaluations>` are listed below.

The requirements for a `<library>` element are:

Requirement 6: `<library>`

- **6**.1 Metadata including a name/title and date of assembly
- **6**.2 An optional `<documentation>`
- **6**.3 A list of evaluations, supporting hyperlinks to data files
- **6**.4 An optional link to another `<library>`. If this link appears, all `<evaluation>` elements not defined in the current `<library>` are inherited from the linked library
- **6**.5 A list of covariances, supporting hyperlinks to data files
- **6**.6 An optional particle database or link to a database.
- **6**.7 An optional set of default representations

The requirements for an `<evaluations>` element are:

Requirement 7: `<evaluations>`

- **7**.1 Metadata including a name/title and date of assembly
- **7**.2 An optional `<documentation>`
- **7**.3 A list of evaluations or links to evaluations (meta or otherwise)
- **7**.4 An optional set of default representations

Discussion point:

It is possible to combine the concepts of a `<library>` and `<evaluations/>`. We felt that separating the concepts treats lists of evaluations, covariances and particles with some degree of parity.

D. Particle and/or material properties database

Task **1**.3 of WPEC Subgroup 38 (SG38) is to define a database hierarchy for handling particle information needed for nuclear reaction evaluations and transport

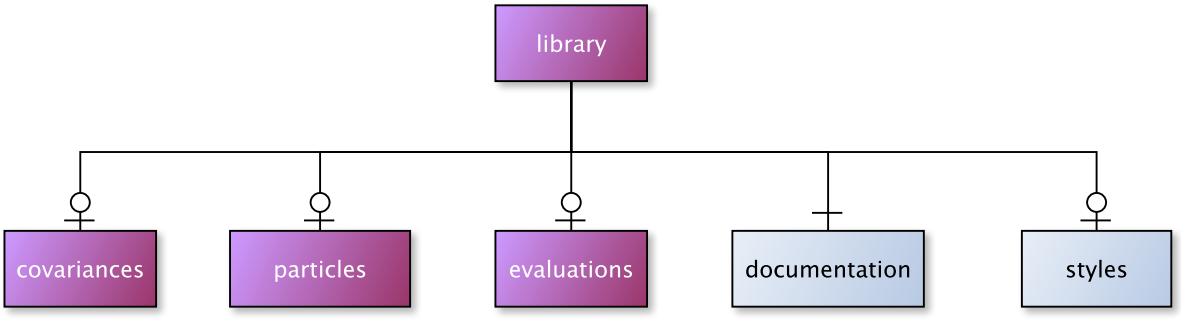


FIG. 2 A `<library/>` groups together all of the parts needed to define a nuclear data (sub) library. This includes evaluations, covariances and/or particle properties. This may also include default representation definitions if needed for a particular data projects.

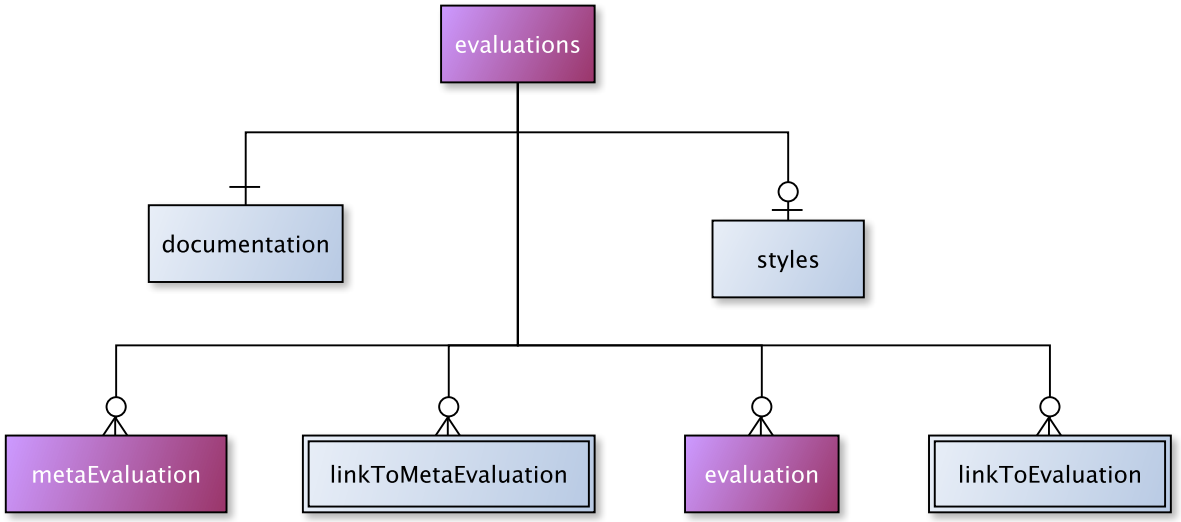


FIG. 3 An `<evaluations/>` element is a listing of evaluations or meta evaluations and any default representation information common to all of the evaluations.

codes. The hierarchy must be general enough to describe relevant particles, including mass, charge, spin and parity, half-life, decay properties, and so on. In a way, the hierarchy encapsulates the needs of the ENDF decay and atomic relaxation sub-libraries as well as the RIPL mass and level tables. Particle databases built with this hierarchy are meant to serve as central locations for particle information that can be linked to from codes and other databases. It is hoped that the final product is general enough for use in other projects besides SG38.

While this is called a "particle database", the definition of particle (as described in Section 2) is very broad and could include materials in the thermal scattering law sub-library as well as atoms and ions. What we refer to as a "particle" includes the projectile and product(s) as well as what the ENDF-6 format refers to as a material or MAT. The database is meant to be general enough to include not only fundamental particles like quarks and leptons, but also composite particles like mesons, baryons, atoms, nuclei and even excited states. Under this definition the list of possible particles becomes quite large. To help organize them the database will need a way of grouping similar particles together into "families" with similar attributes.

One important function of the particle database will be to provide an easy way for codes and external databases to look up any particle stored inside. In order to make this access as simple as possible, the database will include a unique name (or "id") for every particle that it stores. Users can then access a specific particle either by providing a full path to it or simply by using its id. Suggestions from experience with GND:

Aliases: A limited number/scope of aliases for commonly used particles, such as "$e-$" for electron or "a" for alpha or "n" for neutron. Also to associate a nuclear level of an isotope with an isomer.

Compounds: `c_String_Describing_Material` can be used to specify say H in ZrH or the phase of the material. Useful for Thermal Scattering Law data.

Elements: `Sym0` (e.g., `Fe0` or `C0`), useful for atomic data

Isotopes: `SymA` where "A" is the isotope's nucleon number (e.g., `Fe56`)

Nuclear levels: `SymA_eN` (e.g., `V51_e1` for the first excited state of ^{51}V or `SymA_c` for continuum).

Excitations of an atom: `Sym0_eN` (e.g., `V0_e1` for the first electronic excited state of $^{\text{nat}}$V or `Sym0_c` for continuum).

SUGGESTED TEST: Test that all particle ids are unique since they are keys and we must avoid key collisions.

The following terms will be used in the following requirements:

- **matter**: Any small, localized object such as an atom, isotope or elementary particle. Every "matter" instance in the particle database is assigned a unique id.
- **Particle**: A small, localized object that can be attributed properties such as mass, charge, spin, parity, half-life and decay properties. This definition is deliberately broad to include nuclei and excited nuclear states as well as fundamental particles like the electron and photon.
- **Particle Family**: A set of particles that can be classified together (examples include leptons, baryons, isotopes and excited nuclear levels). Particle families may add additional properties that must be defined for all their particles (i.e. lepton number for all leptons, and level energy for all excited nuclear levels).
- **Particle Group**: A particle group is a set containing related particles so that they all appear together within a particle database hierarchy. This is useful when a set of particles have properties in common: the common properties are defined in the particle group, and inherited by all particles in that group. The excited levels of an isotope are an example: they share properties like the proton count Z and the mass number A.
- **id**: A string used to identify and refer to a particle. Each particle in the database must have a unique id.
- **pid**: String used to refer to a particle by id. For example, an alias for the first metastable state in Am242 would have id=`Am242_m1` and pid=`Am242_e2`.
- **qualifier**: Extra key that may be appended to a pid to give additional information about a specific instance of that particle.

The following list of requirements is a summary of the larger discussion of requirements and specifications in the "Requirements and specifications for a particle database" document (WPEC Subgroup 38, 2015a). The key words "shall" and "should" will be used to differentiate between requirements and recommendations:

Requirement 8: Particles

8.1 Each "particle" and "particleGroup" in the database shall have a unique id used to identify and refer to it. Only these objects shall have ids (for example, no id is given to a mass or spin).

8.2 A naming convention for ids shall be defined. Creating and adhering to a naming convention makes comparing and merging different particle databases simpler. If a user wishes to refer to a particle by a name that is not consistent with the naming scheme, they can still do so by defining an "alias" for that particle (see requirement **8**.10).

8.3 Every particle shall contain at least the following properties: mass, charge, spin, parity and half-life. However, some of these properties may be inherited from higher in the hierarchy rather than being listed explicitly (see requirement **8**.5).

8.4 The database shall support storing uncertainties with each particle property. Multiple types of uncertainty should be supported, including central values with uncertainty (for example, mass = 54.938 ± 0.729 amu), upper/lower uncertainty intervals (e.g., mass = 0.03 eV/c^2 - 0.02 + 0.06), and lists of multiple possible assignments (e.g., spin = 3/2, 5/2 or 7/2). If multiple assignments are listed, the database shall assign one as the 'recommended' value.

8.5 The database shall use nesting and inheritance where possible to reduce redundancy by grouping similar particles together. For example, the database should support grouping isotopes with the same atomic number together inside an element, such that all isotopes inherit the same atomic number Z from the element. It should also support grouping excited nuclear states of the same isotope together, such that the mass of each excited state can be computed from the ground-state mass and excitation energy.

8.6 The database shall support defining "families"

to classify similar particles. Each particle family may have additional required data elements (beyond the list in requirement **8**.3). For example, a "nuclearLevel" family should be defined, where each nuclear level requires a level energy and level index in addition to charge, spin, parity, etc.

8.7 The database shall support storing decay properties for unstable particles. Decays are organized into decay channels, which consist of a probability and a list of products. Each product may also have an associated uncertainty. For example, if an excited nuclear level can γ-decay to two different levels, the decay can be stored either as two different decay channels that each contain a probability and one gamma product, or as a single decay channel with two gamma products and their probabilities.

8.8 The database shall support storing documentation sections at multiple levels, down to the level of an individual particle property like mass or spin.

8.9 The database shall support a bibliography section. Each item in the bibliography shall include a unique citation label that can be used to refer to it from any documentation section.

8.10 The database shall support a mechanism for defining aliases for particles. For example, the id `Am242_m1` could be an alias for `Am242_e2`. Alias definitions shall be permitted at various places within the particle database.

8.11 When linking to a particle database, user codes shall be permitted to add extra information about a particle by using "qualifier" keys. For example, a qualifier key may be defined to describe the electron configuration of an atom following a photo-atomic reaction.

Discussion point:

Should the database support flags like "firm", "tentative", etc. to indicate confidence in an assignment? Those could be stored in documentation sections, but if user codes need to know which assignments are tentative they shouldn't need to parse documentation to find out.

Discussion point:

Regarding requirement number **8**.7: There are several ways of handling decay radiation, so this requirement needs a bit more discussion.

Discussion point:

In GND, both reaction data and decay data are sorted into channels, where every channel has a different list of outgoing products. If that example is followed in the particle database, then decays to different excited states in the daughter nucleus would each have their own decay channel. Decay spectra are then reconstructed by looking up each daughter product, checking their decay properties, and summing over all products. For example, a nuclear excited state that can decay via two different gammas would be stored as:

```
<nuclearLevel id="C12_e8" index="8">
  ...
  <decays>
    <decay mode="gamma" probability="0.87">
      <product pid="gamma"/>
      <product pid="C12_e0"/></decay>
    <decay mode="gamma" probability="0.13">
      <product pid="gamma"/>
      <product pid="C12_e1"/></decay>
  </decays></nuclearLevel>
```

The second decay mode leaves C12 in an excited state. Decay properties of that state (`C12_e1`) can be found by looking it up in the database. That particle would contain information about another gamma decay that leads to the ground state and emits a 4.4389 MeV gamma. Gamma energies do not need to be explicitly stored, since they can be calculated from the initial and final level energies. This organization is useful for presenting detailed decay spectra that may proceed through multiple paths. It is less useful when the details are uncertain, such as when a decay passes through unknown short-lived intermediate states on its way to a longer-lived product.

Other databases follow a different organization. For example, the ENDF decay sub-library starts by listing the possible decay modes (as in "$\beta+$", "electron capture", etc.), and then lists the decay radiation (including both discrete and continuous spectra) emitted via all decay modes, potentially including radiation emitted not by the parent but by a short-lived intermediate state. In this case, the `C12_e8` decay scheme would become:

```
<nuclearLevel id="C12_e8" levelIndex="8">
  ...
  <decays>
    <decay mode="gamma" probability="1">
      <product pid="gamma" probability="0.87">
        <energy value="12.71" unit="MeV"/>
      </product>
      <product pid="gamma" probability="0.13">
```

```
      <energy value="8.271" unit="MeV"/>
    </product>
    <product pid="gamma" probability="0.13">
      <energy value="4.4389" unit="MeV"/>
    </product>
    <product pid="C12_e0" probability="1.0"/>
  </decay></decays></nuclearLevel>
```

This is a suitable way of storing experimental decay spectra, since experiments are actually measuring the transitions rather than the particles that emitted those transitions. It is also useful when searching through measured decay radiation to find what parent likely emitted the radiation. On the other hand, it obscures the fact that the 8.271 MeV and 4.4389 MeV gammas occur in coincidence. This style also leads to possible redundancy and discrepancies if more than one parent particle can decay through the same intermediate states, since the same gamma products may also be emitted as part of the decay cascade from a higher level.

Discussion point:

There are some special cases that also need to be considered. For example, how should correlated decays be handled? Example: angular correlations between Ni60 gammas following Co60 β-decay. These angular correlations show up because the two M2 gamma transitions happen back-to-back, meaning that the second decay is not truly independent of the first. Is it sufficient to store the fact that each transition is M2, and require user codes to calculate the angular correlations if needed? Or, do we need some explicit markup indicating that these decays are correlated?

Discussion point:

Another question: current decay databases use the internal conversion coefficient (ICC) when a decay can proceed either through gamma emission or through internal conversion. Should we continue to support that convention (grouping all "electromagnetic" decays together and then using a coefficient to denote relative weights), or should we break them out into separate decay channels?

Another potentially useful convention is to store branching ratios instead of probabilities: each decay mode has a branching ratio BR = (probability of this decay) / (probability of most likely decay). Should we continue to support that convention, or always just store the decay probabilities?

Discussion point:

Should the requirements include a way to describe how "complete" a level scheme is (i.e., how many levels starting from the ground state have firm spin/-parity and energy assignments)? Also, should there be a way of specifying rotational bands for excited nuclear levels?

Discussion point:

At present there are no plans to include the Monte-Carlo Particle Identifiers (PID) from the Particle Data Group (Olive, 2014). That said, the PID's are easy to add to the particle specifications. Furthermore, the Particle Data Group are aware of the project to modernize the ENDF format and they are interested in its results.

E. Documentation

The documentation for an evaluation or part of an evaluation is in a way the most essential piece of information. From it, we must be able to determine who performed the evaluation and how they did it. This is essential both for attributing credit (and blame ;)) and for debugging problems in an evaluation. This also aids in addressing requirement **1**.6. Figure 4 illustrates the required contents of a documentation markup. We now discuss the elements of this markup before listing the requirements for a `<documentation>` element.

Each part of an evaluated data file may be evaluated separately, creating a "frankenevaluation" made up of stitched together evaluation parts. Therefore, we must allow `<documentation>` elements at many different levels in our data hierarchy. Furthermore, as this `<documentation>` element details a subset of information in an evaluation, the `<documentation>` element should contain both version tracking information for that subset as well as an unique identifier to reference the subset of information. The unique identifier should be a valid Digital Object Identifier (DOI). A DOI is a character string that uniquely identifies a piece of information and is associated with the URL where the information may be found. We must have a place for a DOI in the hierarchy because recent initiatives from both the United States and the European Union have imposed an additional legal requirement on scientific data:

- European Union now issues Digital Object Identifiers (DOI) on documents produced with EU funding.
- In the United States, U.S. DOE Scientific and Technical Information Program (STIP), a collaboration working to increase the availability and trans-

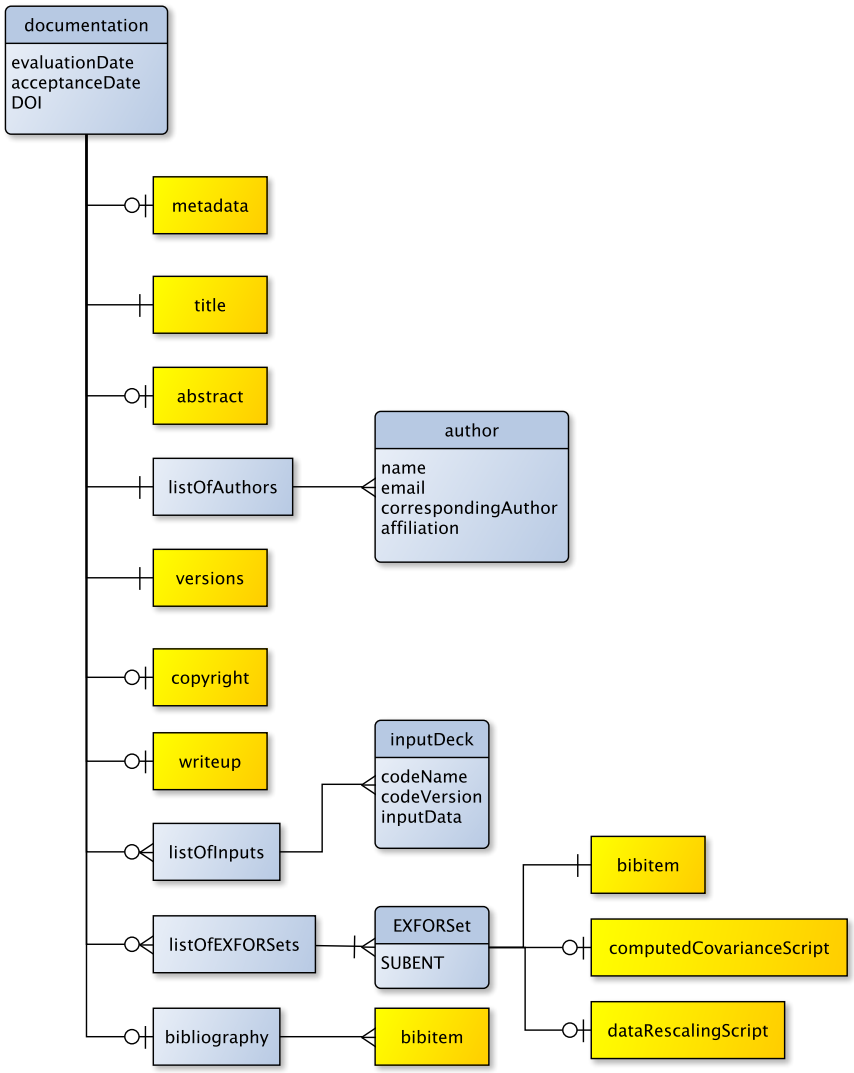


FIG. 4 Basic structure of a `<documentation>` element.

parency of the results of DOE-funded research and development, is working to make US generated data available through Office of Science and Technological Information (OSTI) (OSTI, 2015) through DataCite.org (DataCite, 2015). As part of this effort, the National Nuclear Data Center is authorized to assign DOIs to new datasets.

Since each node in the tree representing the data hierarchy can have its own URL, each `<documentation>` element should be allowed to have its own Digital Object Identifier (DOI).

We illustrate how DOIs can be used to reference independent parts of an evaluation in Figure 5. In this figure, each box represents one discrete part of a schematic evaluation. The coloration of the boxes is meant to represent the independent contribution from different publications, all "green" data come from the document with DOI #0, all "cyan" data come from the document with DOI #2, etc.

Moving down from the top element in figure 4, we note the `<metadata>` element. This element is currently undefined but can be adapted for individual data projects needs. For example, if an evaluation is meant to be indexed in the Nuclear Science References (NSR) database (Pritychenko, 2011), NSR keywords can be placed here. Alternatively, keywords for web searching may be placed here.

The elements `<title>`, `<abstract>`, and `<listOfAuthors>` are clear. The `<versions>` and `<copyright>` elements are other data project specific markups. The `<versions>` element allows a data project to store its own version information in whatever form the data project requires. The `<copyright>` element allows the data project to store the copyright notices for

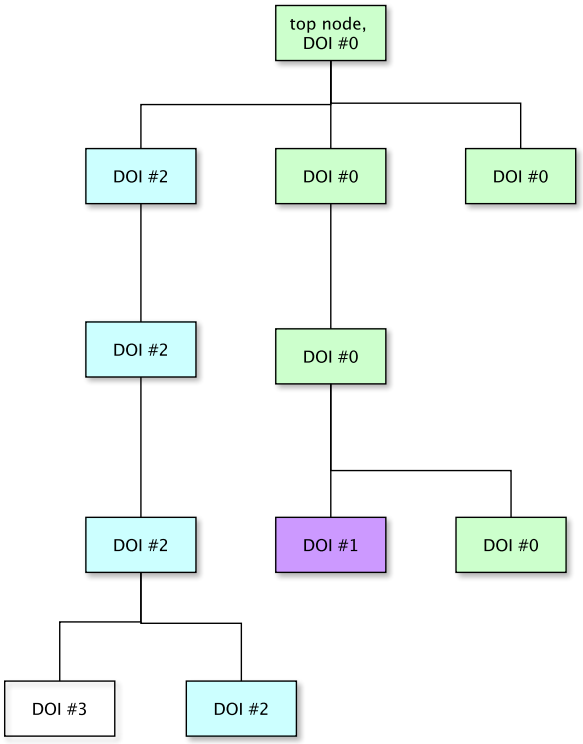


FIG. 5 Cartoon illustrating the construction and documentation of a "frankenevaluation", namely an evaluation stitched together with parts from other evaluations. Each colored node represents an independent contribution of the full evaluation and all contributions are assembled together to form the complete evaluation. At the uppermost node of each set of colored nodes, the evaluator should have a `<documentation>` element and corresponding DOI.

certain parts of their evaluation.

> **Discussion point:**
> It was suggested that a "model only" evaluation be flagged with some form of metadata. However, the form of such a flag is not agreed upon.

It is desirable to have enough detail in the documentation to reconstruct the evaluated data and covariances exactly as the evaluator has produced it (complete with renormalizations of fitted data, etc.). Therefore we require markups for `<writeup>`, `<listOfInputs>` and `<listOfEXFORSets>`. The writeup itself is given in the `<writeup>` element and should be essentially free text (supporting internal formatting such as HTML or LaTeX). The `<listOfInputs>` element allows the evaluator to store say their TALYS or EMPIRE input decks within the evaluation. The `<listOfEXFORSets` is, as the name implies, the list of EXFOR datasets used in the evaluation. The IAEA's EXFOR web application allows both on-the-fly data renormalization (example in Figure 6) and covariance generation (example in Figure 7). Therefore, both require locations in the hierarchy.

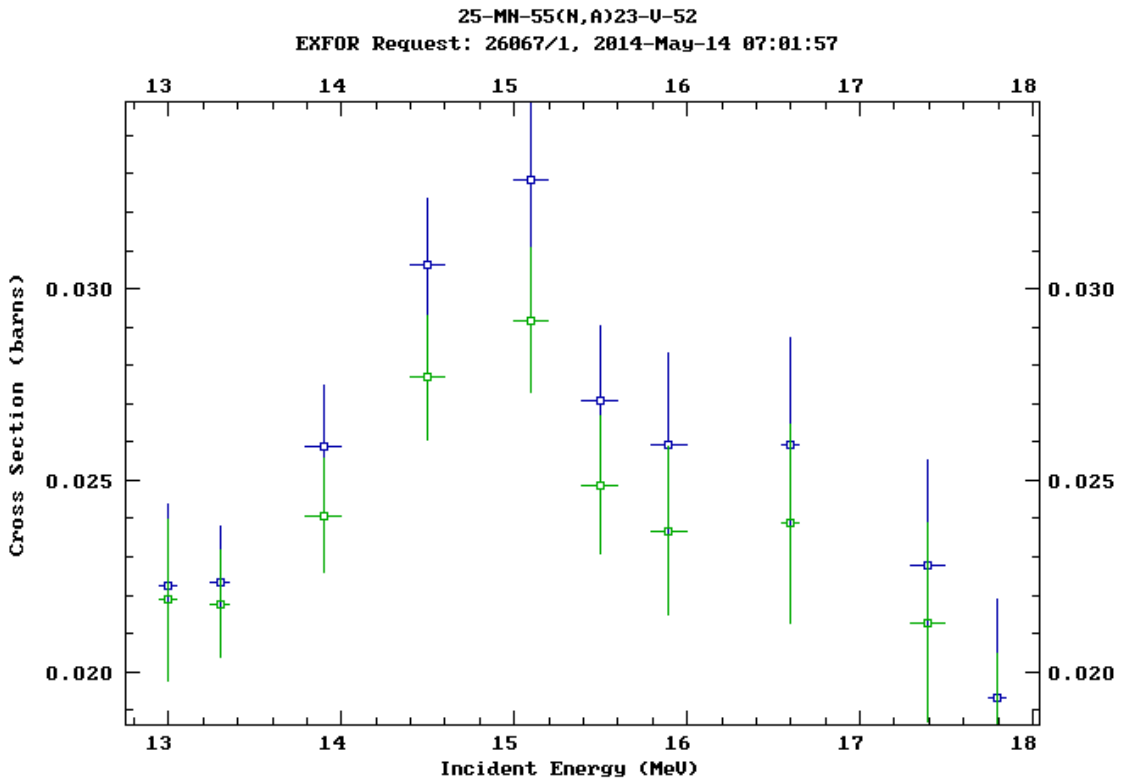
Finally, both `<EXFORSet>`s and the writeup bibliography (in the `<bibliography>` element) require a markup for bibliographic references. The easiest solution here is to adopt an accepted format such as BibTeXML.

> **Discussion point:**
> How should a bibliographic entry be stored?
>
> > **Resolution:**
> > In principal it should be shared with EXFOR. BibTeXML (Gundersen, 2015) was suggested as a viable choice.

The requirements for documentation make use of the `<text>` low level structure discussed below in Requirements **15** and the `<link>` structure discussed in Requirements **16**.

Requirement 9: Documentation

- **9**.1 The data structure shall have metadata element.
- **9**.2 The data structure shall have a markup for the DOI (likely including a `link` attribute)
- **9**.3 The data structure shall have title element (using a `<text>` element)
- **9**.4 Have data structure for the evaluation date (authors generate this), this is a legacy ENDF requirement.
- **9**.5 The data structure shall have a markup for the library acceptance date (library maintainers generate this), this is a legacy ENDF requirement.
- **9**.6 The data structure shall have an abstract element (optional, using a `<text>` element)
- **9**.7 The data structure shall have a markup for authors (names, affiliation, email, etc.). The corresponding author must be denoted.
- **9**.8 The data structure shall have a markup for storing the input decks from codes used by the evaluators to prepare the evaluation (for example `<listOfInputs>` and `<inputDeck>` elements)
- **9**.9 The `<inputDeck>` element should allow free text and therefore must be escaped.
- **9**.10 The data structure shall have a markup for the evaluation version
- **9**.11 The data structure shall have a copyright element (for optional copyright notices).
- **9**.12 The data structure shall have a markup for referencing the EXFOR datasets used in the evaluation. This markup should include a link to the original data and details of any data rescal-




```
30581004  x4u:20090506  #1980 Zupranska
 #Reaction: 25-MN-55(N,A)23-V-52,,SIG
 #Monitor:  26-FE-56(N,P)25-MN-56,,SIG
 #m0: {20377002,H.LISKIEN+,J,JNE/AB,19,73,196502} £ fe56np;#old monit-ref
 m0: exfor$20377002_fe56np;     #old monitor(energy) in EXFOR
 m1: recom$fe56np;              #new monitor(energy)
 dy=dy/y;                       #to rel. uncertainties----
 y=y/m0*m1;                     #renormalized CS
 dy=(dy**2-dm0**2+dm1**2)**0.5;#replace monitor uncertainties
 dy=dy*y;                       #to abs. uncertainties
```

FIG. 6 (Upper panel) Sample rescaling of the ^{55}Mn(n,α) from Viktor Zerkin's EXFOR-Web (EXFOR, 2015) on the IAEA nuclear data services website (IAEA, 2015). (Lower panel) Script used to perform the rescaling of the ^{55}Mn(n,α).

ing or evaluator generated covariances. These can be done using V. Zerkin's EXFORWeb tools (EXFOR, 2015). Examples are shown in Figs. 7–6.

9.13 The data structure shall have an element to store the writeup itself (using a `<text>` element)

9.14 The data structure shall have markup for the bibliography.

Discussion point:

Several users have suggested adding a "key data table" for validation purposes in the introduction. For incident neutrons, this table could include Maxwellian averaged cross sections at room temperature, resonance integrals, the cross sections evaluated at 14 MeV and Westcott factors. For charged particle reactions, such a table would include Maxwellian averaged cross sections at astrophysically interesting temperatures. It is unclear what such a table might contain for other ENDF sublibraries (e.g. thermal neutron scattering). Such a table should use a special markup so that the data in the table can be aggregated and used in global library validation.

```
<?xml version="1.0" encoding="UTF-8" ?>
<CovRecipe  DatasetID="E1922008"
  Created="2014-05-14 06:57:54"
  Software="Web-tool:2012.03.16">
  <Subent id="E1922008" x4upd="20060411" />
  <Reacode code="78-PT-0(P,X)79-AU-196,,SIG" />
  <defineErr name="Statistical" type="C5" cmd="Setup" src="Total" koeff="0.5" ></defineErr>
  <defineErr name="myErr-1" type="myErr" cmd="Setup" src="Systematic" koeff="0.5" ></defineErr>
  <defineErr name="myErr-2" type="myErr" cmd="Setup" src="Systematic" koeff="0.5" ></defineErr>
  <setCompressFactor Factor="2" lx0="58" lx1="29" />
  <addCovarFraction errName="Statistical" errType="C5" corrType="SERC" fracType="Uncorrelated" >
  </addCovarFraction>
  <addCovarFraction errName="myErr-1" errType="myErr" corrType="LERC" fracType="Fully-correlated" >
  </addCovarFraction>
  <addCovarFraction errName="myErr-2" errType="myErr" corrType="MERC" fracType="MERC-correlated"
                    MercType="Lin" MercInterval="0.5" MercEnMin="8e6" MercEnMax="6.94e7" EnUnits="eV" >
  </addCovarFraction>
<myStamp>EXFOR-Web-Covariance-Recipe, V.Zerkin, IAEA-NDS, 2012-03-19.</myStamp>
</CovRecipe>
```

FIG. 7 Sample covariance generation script from Viktor Zerkin's EXFOR-Web (EXFOR, 2015) on the IAEA nuclear data services website (IAEA, 2015).

F. Representations

An evaluation may contain one or more variations of processed or derived data in addition to the original evaluated data. Most data in an evaluation are processed with a common set of inputs. Instead of listing each input set for each processed data, it is less redundant and more efficient to specify the input set in one location, herein called a representation, and use a unique name to reference that representation where it is used. For evaluated or processed data, the representation includes, but is not limited to, target temperature, group structure (in energy and angle) and particle flux weights. Each representation should be uniquely named so that it can be referenced in the lower level data structures.

In GND version 1.7, the element `<styles>` is used to store the representation data. Figure 8 is a sample listing of the `<styles>` element from GND version 1.7.

The items to denote in a `<representations>` element are at minimum:

Requirement 10: `<representations>`

- **10**.1 Shall provide a key to refer to a representation or group of representations.
- **10**.2 Shall provide a mechanism for denoting on a per–transportable particle basis, energy group structures.
- **10**.3 Shall provide a mechanism for denoting on a per–transportable particle basis, angular group structures.
- **10**.4 Shall provide a location for storing the Doppler-broadened target temperature
- **10**.5 Shall provide a mechanism for uniquely naming and referring to representation information.
- **10**.6 Shall provide a mechanism for storing fluxes for flux weighting of projectiles.
- **10**.7 Should provide a way to name a representation or group of representations.
- **10**.8 Should provide a cut-off time to separate prompt and delayed particle emission and to specify the upper time limit of consideration. This is useful for describing decay data, fission

```
<styles>
  <evaluated name="eval" temperature="300 K"/>
  <linearized name="lin1" temperature="300 K"/>
  <linearized name="lin2" temperature="100 meV/k"/>
  <linearized name="lin3" temperature="10 eV/k"/>
  <multiGroup name="gLLNL1" temperature="300 K">
      <flux ... > ... </flux> <!-- flux id 12 -->
      <groupBoundaries pid="n" ... >
      ...
      </groupBoundaries>
      <groupBoundaries pid="H1" ... >
      ...
      </groupBoundaries></multiGroup>
  <multiGroup name="gLLNL2" temperature="300 K">
      <flux ... > ...  </flux> <!-- flux id 43 -->
      <groupBoundaries pid="n" ... >
      ...
      </groupBoundaries>
      <groupBoundaries pid="H1" ... >
      ...
      </groupBoundaries></multiGroup></styles>
```

FIG. 8 Sample listing of the `<styles>` element from GND version 1.7

product yields and delayed neutron emission after fission.

10.9 Should provide a scheme to store the code inputs for the processing code used to generate data in a given representation.

Other data may be added as the needs arise.

Discussion point:

As `<representations>` may be provided for an entire library or on a per-evaluation basis, we must establish precedence rules.

Discussion point:

This scheme replaces to some degree the need to provide a list of links for every derived data item. Now only a representation key needs to be provided in a piece of derived data.

G. What data are derived from what other data?

As a corollary to main Requirement **2**.2, we require a mechanism to specify what data are "original" and what are derived. To accommodate this, we must allow the storing of the original and derived data, sometimes at the same level in the hierarchy. Derived data must point back to the original data with some key or a `link`. We note that while some data are derived from only one other element, some data are derived from several disparate pieces of information within an evaluation.

There are many cases where a capability to link original and derived data would be useful. Here are a few that come to mind at the time of this writing:

- Doppler broadened data at a temperature $T > 0$°K should link to the 0°K data.
- Grouped and pointwise data
- Angular distributions converted between pointwise angular tables and Legendre moments
- Any (and all) parameterized data converted to pointwise
- Average energy deposited, average forward momentum deposited and KERMA factors are all derived using product distribution data and energy balance of all the particles emitted in a reaction
- Changes in interpolation schemes (e.g., log-log to lin-lin)
- Resonance data converted to pointwise
- Resonances with smooth backgrounds (this is allowed in ENDF, and we argue below that it should be deprecated in the new hierarchy).
- Cross section (and other data) may have uncertainty data and should be associated with any correlation matrices. Grouped or other representations of the data need to point back to both the uncertainty data and the mean value data.
- Monte Carlo realizations of a data set should point to the mean value and the associated covariance

Fig. 9 illustrates one scheme that achieves this for a mid-level container. The top level of the mid-level container may contain within it several representations of the underlying data. In this cartoon, the different representations are stored in a `<listOfDataRepresentations>` element. The mid-level container must then have a link to the actual dataset in `<listOfDataRepresentations>` that stores the original or "official" data. This simplifies navigation since we do not have to query each data to determine which version are the original data. Each contained dataset has a flag to denote whether the set is "official" or not and all derived data contain a link to the data from which it was derived from. In this example, set #2 is derived from set #1 (it might be from a group collapse for example) and set #1 is derived from set #0, the original set (set #1 may be the grouped version of set #0). We comment that default representations can be stored in the upper level `<representations>` element.

This concept was pioneered in GND. In GND, the different versions of the same data are each encapsulated within a `<form>` element. The `form` element either is the data container itself or the lowest level of the top-level hierarchy before encountering the actual containers holding the data. GND `link`s derived data to the original data by indicating how data are derived in the `<styles>` section.

Figure 10 shows another example of how original and derived data can be handled. In this figure, we show the original cross section given as the mean values of the cross section and a covariance matrix. From that, one can derive a second representation of a cross section with uncertainties as well as a correlation matrix.

We now summarize the general requirements for a mid-level container that supports linking between original and derived data:

Requirement 11: Linking between original and derived data

11.1 A data containing element shall have a `<listOfDataRepresentations>` that contain the different version of the data. Depending on how this is finally implemented, this may be the top-level element itself (like

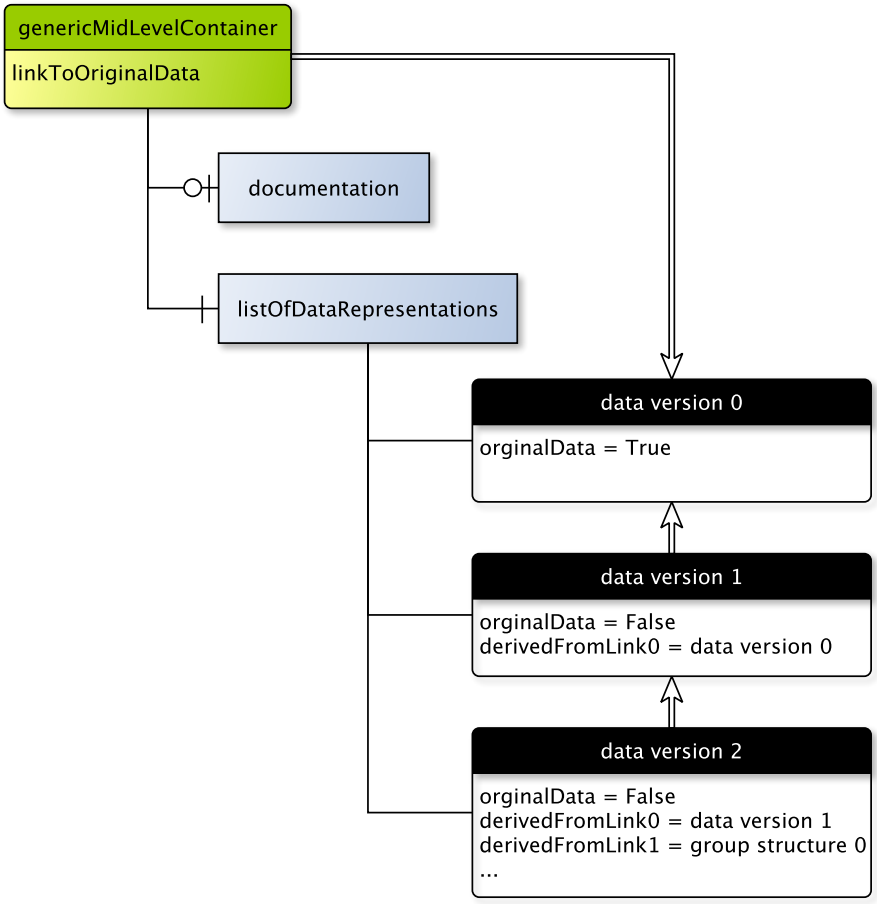


FIG. 9 Sketch of a generic mid-level container. Note the use of links to represent what data were derived from what other data and to denote which data sets were the original. This motif is used repeatedly throughout the hierarchy. Note that the `originalData` and `<derivedFrom>` are redundant in this example; we really only need a `<derivedFrom>` element to denote what is derived and what is original.

`<crossSection>`) or a list container within the top-level element.

11.2 A data containing element shall have one specific implementation of an element with low-level data plus additional attributes.

11.3 Derived data shall have a mechanism to denote what pieces of information were used to create the data in question. This may be accomplished with a `<listOfDerivedFromLinks>` with one or more `<derivedFrom>` links to the original data or similar scheme. Alternatively, this might be accomplished in the `<representations>` section and inherited by all data using each `<representation>`.

11.4 Derived data should be placed as close to the original data as possible.

Discussion point:

There are times when data which are traditionally viewed as "derived" data actually are evaluated and placed in a file. This is sometimes the case with radiation damage data (PKA, DPA or KERMA) or with beta-delayed data after fission. These data must be flagged as the "evaluated" version of the data even though it might live in a container that might suggest otherwise such as a possible `<derivedReactions>` upper level element.

H. Prototyping functions

We foresee the possible need to try out new interpolation schemes or functional forms in various data. In the development of the legacy ENDF format, new functional forms and formats are proposed at the annual CSEWG meeting and the ENDF formats committee approves/re-

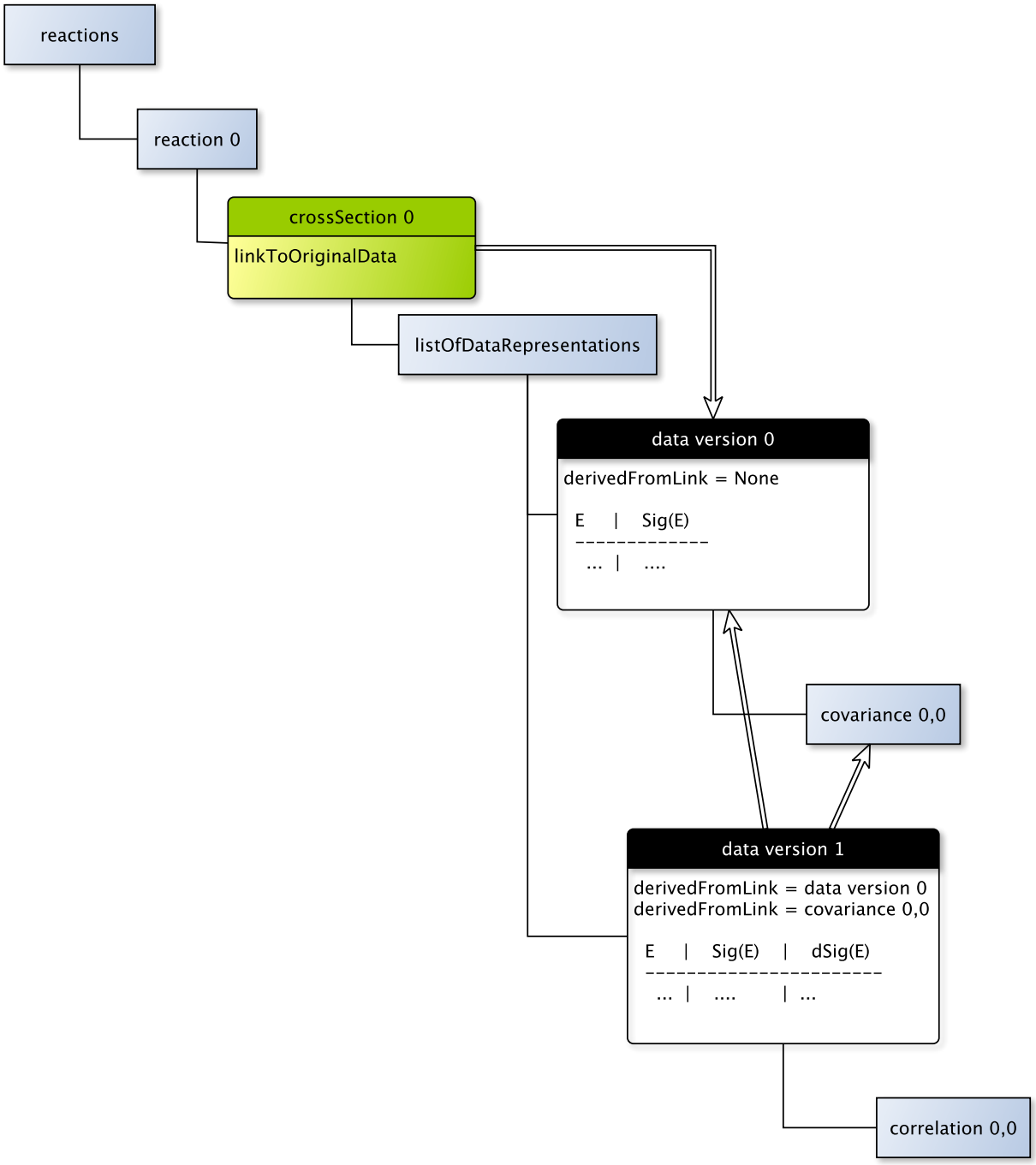


FIG. 10 An illustration of uncertainty in a data set, but derived from original data and a covariance. The coupling between reaction data and the corresponding covariance is handled through use of hyperlinks.

jects them. However, in this approach, there is little guidance other than the ENDF format proposal documents that tell one how to implement a specific new capability. Here we propose a light weight scheme to document new functional forms and interpolation schemes in pre-production evaluations. This set of instructions can guide the developer of a processing code how to implement a new functional form and provide the requisite documentation. In Fig. 11, we outline the concept of a `<functionDef>`. We note that in an evaluation which uses an `<functionDef>` may have data that uses the scheme described by the `<functionDef>`, therefore that data should link to the `<functionDef>` in some fashion. Also, the list of all defined `<functionDef>` should be an

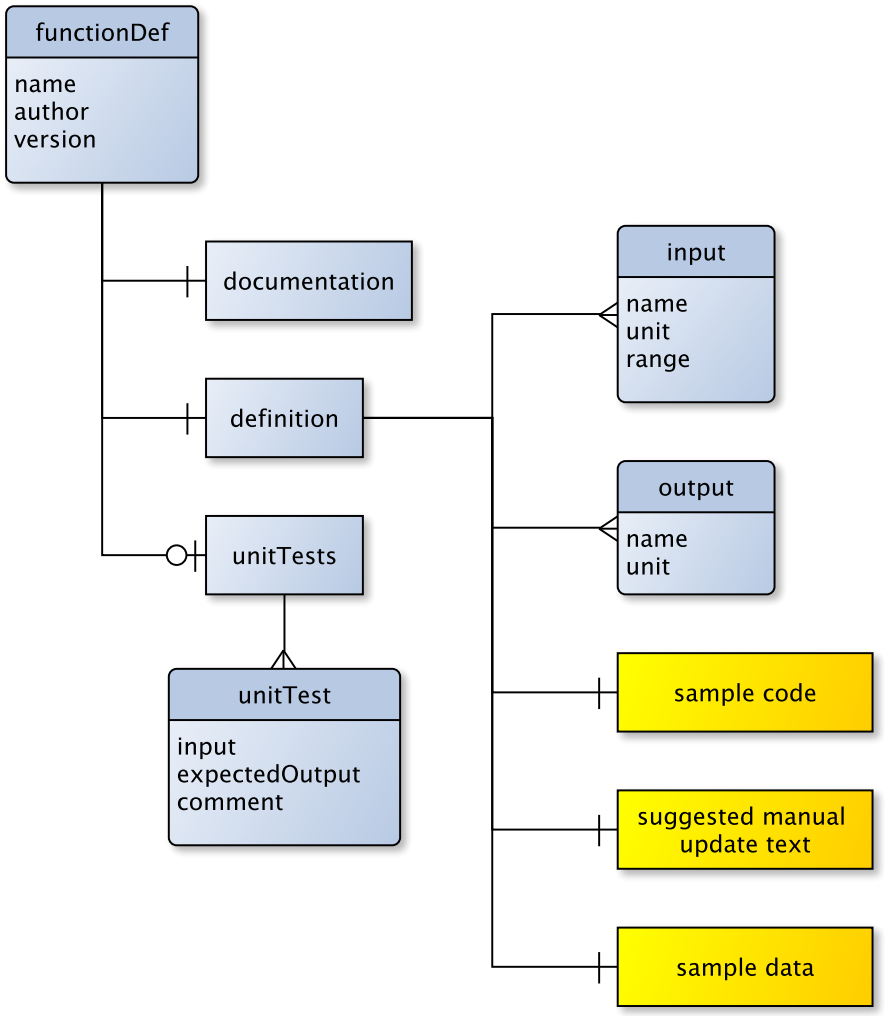


FIG. 11 The `<functionDef>` Outline of the `functionDef` scheme for prototyping new functional forms and interpolation schemes.

optional list of elements dangling off an `<evaluation>` or equivalent document root node.

Requirement 12: `<functionDef>`

12.1 A top level node such as an `<evaluation>` element that optionally contains a `<listOfFunctionDefs>` containing the actual `<functionDef>`'s.

12.2 The `<functionDef>` must have attributes or sub-elements that define the name, the author(s) and version of the definition.

12.3 There are three components that are crucial for implementing a `<functionDef>`:

12.3.1 Detailed documentation in a required `<documentation>` element.

12.3.2 The definition of the `<functionDef>` in a `<definition>` element. This includes lists of expected inputs (including name, unit and range) and outputs (again including name, unit and range), sample code showing how the new feature is implemented, sample data generated in the new scheme and suggested text for the successor to the ENDF format manual.

12.3.3 Unit tests so that the implementation `<functionDef>` can be tested against the authors' expected results. The unit tests, in `<unitTest>` elements, contain the inputs, expected outputs and any documentation/comment describing each test.

Discussion point:

The `<applicationData>` area of the hierarchy (see

requirement **18**.18) may also be used by individual institutions to prototype new data and formats. However, `<applicationData>` has no "rules" regarding its use so it may be less useful for sharing new data types/formats.

I. Required low-level containers

In a hierarchical structure, nodes most often contain another collection of nodes. However, at some point one reaches the bottom. In the case of a structure used to store scientific data, the lowest-level containers must store numeric data. This includes, but is not limited to, integer, fractional, real, and complex values that describe, for example, interaction probabilities, angular states, decay constants, barrier penetrabilities, etc. or mathematical functions in a tabulated form.

Consideration must be given to the issue of reading and writing data to memory. The ENDF format is a marvel of efficiency considering the 80 character line length inherited from its punch card origins. Each line knows its exact place in a file, its material, data type and reaction and stores up to six associated values. However, to maximize the significant figures available for its eleven character numeric values, ENDF allows one to drop the "e" from an exponential leaving, for example, "1.2345+7" in place of "1.2345e+07". Given the strong Fortran roots of our community and Fortran's understanding and acceptance of this format, it provides the broadest compromise. However, it comes at the expense that today's modern languages – C, C++, Perl, Python, etc. – must parse these values ad-hoc at considerable expense in extra read times. Any modern data structure must store numeric values only in the widest adopted standards.

In practice all data – scalar, vector or multi-dimensional – can be serialized and stored in a contiguous block of memory, that is a vector of numbers. In binary form, these data tend to be the same in use, for example active RAM, or archived, for example on disk. An enormous time saving is available for such data if their movement is done by large block reads and writes that forego the need of conversions. Even in textual form, transformation to and from text of a sequence of numbers may be done much faster when it does not contain extraneous text. As the bulk of the data we intend to store is numeric, providing a "lowest-level" container that is strictly a vector of numbers can provide enormous time savings.

Unfortunately, nodes tend to be expensive to create in practice. This has implications for scalar data. While a low-level vector array container provides clarity and space savings for sets of numbers, it may not be necessary for a scalar quantity. In fact, there are times when it can be a significant waste of space. Consider two XML examples of storing the mass of a nuclei: "`<mass array="true" size="1" id="U235"> 235.043 </mass>`" versus "`<mass id="U235" value="235.043" />`". The first example adds a minimum of 20 extra bytes for each scalar quantity and actually obscures the value with unnecessary attributes. For other storage formats, for example HDF5, this can be even worse. However, there is a clear need to recognize data, as opposed to meta-data, that are stored as an attribute. This argues for a clear universal attribute name to indicate the scalar data value.

Given this, we can write some general requirements for low-level containers. The requirements below are a summary of the larger discussion of requirements and specifications in the "General–Purpose Data Containers for Science and Engineering" document (WPEC Subgroup 38, 2015b).

Requirement 13: General low-level

- **13**.1 General-purpose data types shall be defined that are compatible with commonly used computer languages as well as their common usage and libraries.
- **13**.2 Containers should make efficient use of computer resources. This is a compromise between:
  - **13**.2.1 efficient use of memory - volatile (e.g., RAM) and non-volatile storage (e.g., disk drive).
  - **13**.2.2 ease of conversion to other forms.
  - **13**.2.3 time to convert to other forms.
- **13**.3 The structures should reduce redundancy.
- **13**.4 Data containers should be designed to be consistent with object-oriented programming. This includes the nesting of data containers when it make sense instead of defining a new sub-container.
- **13**.5 Data should be stored in the smallest possible container, but no smaller
  - **13**.5.1 Singular scalar values may be stored as attributes where space savings is significant
  - **13**.5.2 Scalar values stored as attributes shall share a universal attribute name
- **13**.6 Nodes storing only numeric data shall be clearly marked for ease of parsing
- **13**.7 Numeric data shall not be stored in text fields whose primary purpose is documentation and shall use only generally accepted forms. Storing data in fields whose primary role is storing non-numeric data should be avoided as much as possible.
- **13**.8 The specification for a data container shall state whether that container is extensible. If the container is extensible, the specification

shall define the process for extending the container.

13.9 A mechanism for storing units and labels for data should be specified for each container type.

Discussion point:

Should scalar values stored as attributes use a common naming convention? In other words, should we require the temperature always to have the attribute name `temperature`?

Resolution:

There is no general rule and we must (re)visit this issue on a case by case basis.

1. General data/functional containers

At its simplest level, all numeric data can be broken into arrays of numbers and meta data describing the mathematical interpretation of the data. These should be grouped together into some kind of general data container. These general data containers serve the purpose of defining the mathematical and functional forms that are themselves basic components of higher level physical data storage containers. This addresses requirement **2**.4.

Although the notation here is slightly different that that in the specifications document for general-purpose data containers (WPEC Subgroup 38, 2015b), we hope the naming is clear enough for the reader to understand the mapping of elements between the two documents.

Requirement 14: List of general purpose elements

14.1 Floats (`float`), integers (`int`) and strings (`string`)

14.2 List or vector (`<list>`), must specify type of object in the list

14.3 Array (`<array>`), must specify dimensions. May be banded, symmetric, etc.

14.4 Table (`<table>`), like a 2-dimensional array, except it may contain non-numeric data and permits columns to have labels, units and data-type information

14.5 Orthogonal function expansion, Legendre polynomials and spherical harmonics being the most obvious

14.6 One-dimensional interpolation tables (`<interp1d>`): interpolation table for univariate data (i.e., x vs. $f(x)$)

14.7 Two-dimensional interpolation tables (`<interp2d>`): interpolation table for bivariate data (i.e., (x, y) vs. $f(x, y)$)

14.8 Three-dimensional interpolation tables (`<interp3d>`): interpolation table for trivariate data (i.e., (x, y, z) vs. $f(x, y, z)$)

14.9 Axis elements (`<axis>`) to specify units, labels, etc.

14.9.1 Specify names of x, y, z, ... axes

14.9.2 Specify units for all axis elements

14.10 Grid elements (`<grid>`) to perform the same role as an `<axis>`, but including bin boundaries. This is for use in specifying group boundaries in derived data or in covariance data.

14.11 Interpolation details

14.11.1 Specify interpolation scheme(s) or group boundaries

14.11.2 If interpolation refers to a probability density function (PDF), we also must specify whether is Normal or Log-Normal (Zerovnik, 2013). This avoids the requirement of adding additional context to denote normalization constraints and helps avoid a source of error.

14.12 Free text (`<text>`) (more on this in subsection II.I.2)

14.13 Hyperlinks (`link` attributes) (more on this in subsection II.I.3)

Discussion point:

Do we need to support complex numbers as well? It might help in the resolved resonances and with atomic reactions. It would also likely complicate specifying and using covariances on complex valued data.

Resolution:

Yes, some variants of resonance formats (both processed and evaluated) require complex numbers. However, these variants are very uncommon and we do not anticipate defining requirements for them at this time. Nevertheless, support for complex numbers should be considered. To support complex numbers in a platform-independent way, the best option may be to store two vectors representing the real and imaginary parts of each complex number.

Discussion point:

It has been suggested by several members of the nuclear data community to include uncertainty directly into elements such as the `<interp1d>` table. This

would make plotting the uncertainty simpler at the expense of introducing an additional data synchronization problem between the mean values and the covariance data.

Resolution:
This idea is adopted since it moves the data and its uncertainty/covariance together. If the covariance or correlation accompanying the uncertainty is not stored with the uncertainty data, a `link` to the covariance or correlation is needed. See subsection II.G.

2. Text

There is a common need for unstructured and partially structured text in evaluation and other documentation. There may be other, less obvious areas. In the documentation, one must store author names and affiliations and do so in non-Latin alphabets. We require full UTF8 support (unicode) so at the very least author names can be written in their own language. This also enables each data project to write their own documentation in their own native language. Since encoding needs to be stated separately from the text itself, a `<text>` element must never be stored in an XML attribute field. Finally, we note that since text may be formatted, the internal `<text>` format must be denoted. Given that the reference format for which we are detailing requirements in will be a subset of the XML markup, we strongly recommend against allowing raw XML for data formatting so as to avoid parsing problems.

Requirement 15: Text elements

15.1 Specify text formatting (HTML, Markdown, etc.). This formatting should not be XML if it can be avoided.
15.2 Allow UTF8 encoding

3. Hyperlinks

Links (`links` are attributes in XML) are an important part of the new data structure and allow the evaluator to refer to other elements within the file or even to elements in external files or databases. Examples of data which use links include:

- Distributions for one reaction product may be treated as the recoil from another product, requiring a link to the other product.
- Production cross sections may be listed as an energy-dependent multiple of another cross section, requiring a link to the other cross section.
- Covariances are occasionally stored in a separate file from the quantities they correlate. Links are necessary to associate the covariance with the correct data.

Because the data are stored hierarchically, the path within a document can be followed straightforwardly. It is useful to think of these paths as similar to paths in a Unix filesystem, but with the top level of a document referred to with a URL.

Requirement 16: Hyperlinks

16.1 The paths may be absolute so that they can refer to external documents or relative so that they can make in-document referrals.
16.2 The URLs of the schema (in the case of an XML version of the data structure) may not be accessible by some computers "behind the fence" so these URLs may be viewed as "placeholders" that can be overridden in specific applications.
16.3 Links must also provide a mechanism to address specific datum within a lowest level container for cases where the lowest level container itself is structured (for example, elements in a matrix or array).

We note that standard xPath syntax does not allow addressing within low-level XML containers (that is, containers containing no other XML markup). Therefore, addressing elements in e.g. a low-level array will require additional XML syntax. Such a capability will prove useful when say varying a cross section at a specific energy. In this case one needs to address a specific point in a low-level container (or range of points) but have no xLink syntax to connect to those points.

One can easily imagine that one is using a nuclear data library on a computer not directly connected to the internet so external links may not be available. In that case, it would be up to the user of the data to remap the URLs to the actual location of the data files on their own computer system. Anything we can do in our API specifications to simplify this task would be appreciated by the user.

Requirement 17: Placeholder hyperlinks

17.1 There should be support for place-holder names in URLs. For example, `<myElement href="$(NUCLEAR_DATA_PATH)/fluxes"/>` where the NUCLEAR_DATA_PATH is defined by a particular institution.

SUGGESTED TEST: Check for valid links for all `xpaths` and `hrefs`.

The extensive use of hyperlinks in this format will lead to at least one new potential source of error: recursive hyperlinks. One can imagine an evaluator saving space by using hyperlinks in lieu of multiple copies of a distribution. But what if the evaluator makes the mistake of never specifying the "main" distribution and only links the distributions back on each other so no distribution is actually ever given?

SUGGESTED TEST: Check for recursive links

III. EVALUATIONS

The top level of data files in all major libraries is the "evaluation", consisting of one target material and one projectile and all the data that goes with the reactions between this pair. This arrangement is familiar to the nuclear data community and should be embraced going forward. That said, we recognize that the name "evaluation" implies that we are only interested in evaluated data, which is far from the truth.

Because of the different kinds of evaluations, what happens below the uppermost node in the hierarchy can differ from sub-library to sub-library. There are four main classes of reaction sub-library that concern us:

- **Thermal scattering law data:** neutrons reacting with such low energy that the de Broglie wavelength of the neutron is too large for the neutron to resolve individual nuclei (in principle other particles such as γ's could do this too)
- **Atomic scattering data:** electron and photon interactions with atoms
- **Nuclear reaction data:** any projectile impinging with enough energy to interact with an atomic nucleus. This collection of data can include resonance data which is arguably different enough from fast reaction data to merit its own discussion.
- **Fission product yield data:** tabulations of fission product yields both directly following a fission event (and prompt neutron and gamma emission) and after beta decaying. These data are arguably reaction data, but their use in radiochemistry, heating and depletion calculations make them important enough to consider separately.

An ENDF-like decay sub-library is discussed in the context of a material properties database.

For the purpose of discussion, we name the top-level element of an "evaluation" `<evaluation>` (see §III.A). In GND, the equivalent concept is called a `<reactionSuite>`. The `<evaluation>` element is expected to be the root XML node of a file. Similarly, when connecting several evaluations together to create an "effective" evaluation, we name the new "effective" evaluation a `<metaEvaluation>` (see §III.B). Finally, as mentioned earlier in this document, lists of `<evaluation>`s and `<metaEvaluation>`s are collected in a `<evaluations>` element.

A. `<evaluation>` elements

In all cases, the data can be arranged using a consistent set of rules. Because of the different optimal representations in the fast and resonance regions, we must distinguish between evaluated data

- broken out by reaction and tabulated one at a time (using the `<reactions>` element) or
- tabulated in a parameterized form such as in the resonances region (using the `<resonances>` element).

We also need spots for covariance data that is not associated with particular reactions (e.g., cross–reaction covariances). Furthermore, we must provide markup that supports transport and other applications through derived transport data markups and a `<representations>` markup for things like default group structures. We mention the list of function definitions that are detailed in section II.H as the `<functionDef>` markup would only rarely be used and used only in "unofficial" evaluations. Finally, at the top level we store a local particle properties database in the `<particles>`. This is meant to override default values given in some external reference database. This top level arrangement is shown in Figure 12.

We comment that an `<evaluation>` element must contain `temperature` and `activationFlag` attributes. For neutrons, Doppler broadening is important to determine effective reaction rates and to get self-shielding corrections. For astrophysical applications, the temperature of the plasma is needed to handle charge screening properly. For activation data, we require an `activationFlag` attribute to signal whether the data in this evaluation is meant for activation or for particle transport. The two applications have very different completeness requirements that, in the XML variation of a data structure, can be enforced by checking against an XSD file.

Requirement 18: `<evaluation>`

- **18**.1 Require one projectile (e.g., "`n`")
- **18**.2 Require one target material (e.g., "`Fe56`")
- **18**.3 Require the versions of the data structure and data format
- **18**.4 Require the library designator (i.e., a name string that says "ENDF/B" and a version string that says "VI.1")
- **18**.5 Optionally a file-wide specification of the Lorentz frame of the incident energy of the projectile
- **18**.6 Optionally support other metadata the library maintainer needs for proper data management.
- **18**.7 Require a `temperature` attribute.
- **18**.8 Require `ELow` and `EHigh` attributes to specify the energy range of validity of this evaluation.
- **18**.9 Require an `activationFlag` attribute to signal whether the data in this evaluation is meant for activation or for particle transport.
- **18**.10 Require a file-wide `<documentation>`

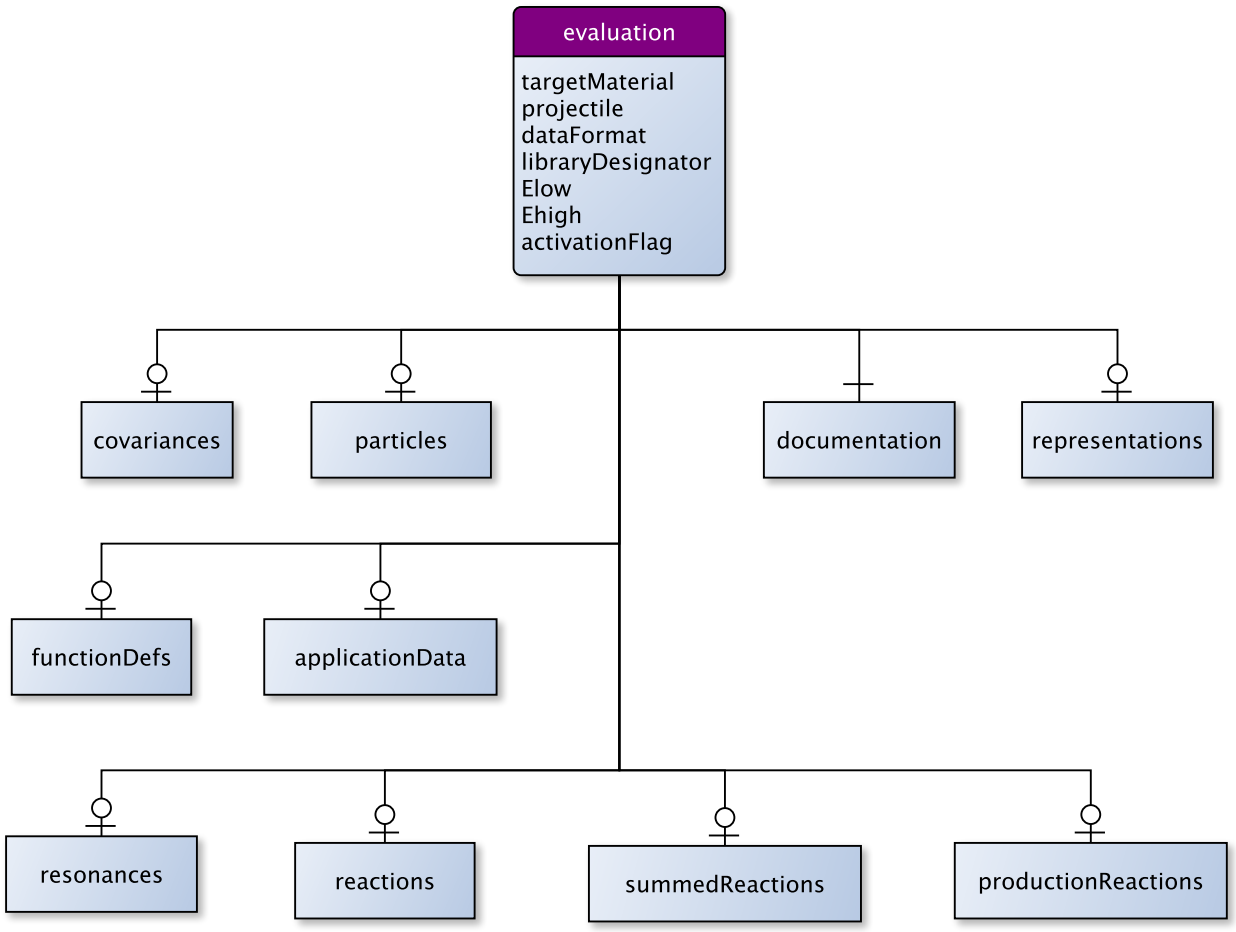


FIG. 12 Top level arrangement of an `<evaluation>` element. Only the documentation element is required in an `<evaluation>`, but `<reactions>`, `<summedReactions>`, `<resonances>`, and `<covariances>` are expected in nearly all (neutron induced) reaction evaluations. The `<representations>`, `<particles>` and `<functionDefs>` elements are used primarily to override or (re)define default behaviors. Finally, the `<applicationData>` element is nearly exclusively for processed data.

18.11 Optionally a material database to override defaults with values local to the evaluation (the `<particles>` element is summarized in §II.D and described in reference (WPEC Subgroup 38, 2015a))
18.12 A place for evaluation–wide default representation information (see the `<representations>` element description II.F)
18.13 Optionally a place for covariance data (see the `<covariances>` section for more detail IV)
18.14 Optionally a `<reactions>` element (more on `<reactions>` in the subsection III.C)
18.15 Optionally a `<resonances>` element (more on `<resonances>` in the subsection III.G)
18.16 Optionally a `<productionReactions>` element (see subsections VI and III.E.1)
18.17 Optionally a `<summedReactions>` element (see III.C.1)
18.18 Optionally a `<applicationData>` element to store application specific data for each institute.

Discussion point:

As the resonances and the fast regions are two distinct physical representations of data in two different energy regions, it might make sense to require that they NOT be together in the same evaluation and that users use the `<metaEvaluation>` markup to combine them. This simplifies bookkeeping and ensures that users understand that they are different things that they must combine themselves. However, this is a change from the ENDF mindset and would complicate translation of the outgoing particle

distributions in the resonance region. Many ENDF forms don't support the calculation of outgoing angular distributions from the resonance parameters so evaluators must provide these tables.

Resolution:
This idea is too much of a change from the ENDF arrangement and will likely lead to confusion.

Discussion point:
Within an evaluation in a particular sub-library, one must ask whether to arrange the data per-energy or per-reaction. For data to be per-energy or energy-major, we mean that all data (reactions, cross sections, distributions, etc.) for one incident energy are collected together in one parent element. For data to be per-reaction or reaction-major, we mean that all data (cross sections, distributions as a function of incident energy) for one reaction are collected together in one parent element. A per-energy arrangement is particularly convenient for Hauser-Feshbach (and other) modeling codes because one normally computes one energy at a time and loops over incident energies to assemble the full evaluation. An energy-major arrangement has certain benefits and drawbacks:

- **Energy-major benefits**
  - Natural output of a reaction model such as EMPIRE or TALYS
  - Energy-major is natural for sampling in Monte Carlo transport: in a collision in a Monte Carlo calculation, one samples between the reaction cross sections for different channels at a fixed incident energy
  - One can see at a glance what channels are open and compete with one another
- **Energy-major drawbacks**
  - Very difficult to plot say a cross section as a function of incident energy
  - Can be difficult to compare to experimental data taken at different energies
  - Although most Monte Carlo transport codes sample on a per-energy basis, the codes are all written assuming a reaction-major arrangement of data and therefore would require major refactoring to reap the benefits of an energy-major arrangement. Besides, if an incident energy falls between two tabulated energies, all sampled quantities must be interpolated.
  - It is difficult to diagnose unphysical discontinuities as a function of incident energy
  - Even a single resonance in a resonance regions span many energies so an energy-major arrangement seriously complicates resonance reconstruction
  - It is hard for deterministic codes to use
  - Not familiar to users as legacy ENDF data are stored with the reaction-major arrangement

Resolution:
The arrangement of data must reflect the users' needs first, not the evaluators. Therefore, we argue that the benefits of an energy-major arrangement do not outweigh the drawbacks and so we recommend maintaining the ENDF-style reaction-major arrangement. However, denoting the energy range of validity of an evaluation coupled with the `<metaEvaluation>` concept would allow an evaluator to achieve the effect of an energy-major arrangement by having only one incident energy in an evaluation. It was suggested that someone develop a tool to combine these one-energy sized evaluations into a complete reaction-major evaluation. This tool would then be reusable for data generated using any Hauser-Feshbach code.

Discussion point:
Do we need an index or directory outlining the evaluation such as what Red Cullen produces with DICTIN (part of PREPRO)?

Resolution:
No, the hierarchical arrangement of data should make this unnecessary

Discussion point:
It was suggested that we allow `<particles>`s at other levels in the hierarchy to, say, allow per-reaction mass of target specifications. This would make it easier to re-use legacy evaluations that perform well in applications but yet use older values of fundamental data.

B. "Meta" evaluations

There is a relatively common need to "glue" together evaluations to make new "effective" or "meta" evaluations. This is often used to connect evaluations from different physical regions or to assemble new reusable materials in input deck specifications. For example:

- In LANL's MCNP code system, the `xsdir` file allows one to connect the thermal neutron scattering data with the neutron nuclear reaction data and even various high energy models such as CEM. See, for example, Figures 14 and 13 .
- The LLNL transport codes AMTRAN and Mercury both allow one to define target macros to describe the material in a zone.
- ORNL's SCALE package contains a pre-built material composition database.
- At the Canadian Nuclear Laboratory, there is another, similar, facility to connect thermal neutron scattering data at different temperatures and even different phases of the target material.

Figure 13 illustrates the need for assembling a "complete" light water evaluation from other evaluations. Figure 14 illustrates the need for assembling an evaluation for projectiles spanning a variety of energy ranges.

There are other uses for being able to connect evaluations together:

- Defining elemental evaluations
- Grouping data on same target, but heated to different temperatures
- Defining generic fission fragments through a weighted average of fission fragment evaluations
- Putting together the parts of a TSL evaluation at fixed temperature, but including all the scatterers.
- Defining common material definitions. This helps answer the question "Which concrete?"

Ideally, these could be shared but rarely are because of the wildly different formats used by various projects. The need for "gluing" together evaluations is so common that we should seriously consider supporting it.

The idea of a `<metaEvaluation>` is straightforward. One uses a set of `<grid>`-like elements to define the grid in some parameter space one wishes to populate with evaluations. The `<grid>`-like element's could be temperature, incident energy, pressure, etc. The `<grid>`-like element defines the boundaries in the parameter space. The `<grid>`-like elements also define the interpolation scheme to be used in that parameter's direction, but in practice the interpolation information will probably be ignored because each project defines their own rules for stepping up in temperature, etc. Figures 17 and 15 provide examples of `<metaEvaluation>`'s for water and natural oxygen.

Requirements for a `<metaEvaluation>` are:

Requirement 19: `<metaEvaluation>`

19.1 An `projectile` attribute to define what projectile this `<metaEvaluation>` is only valid for (say TSL+fast gluing only for neutrons).
19.2 `<axis>` elements to define the grid in which the evaluations will be inserted
19.3 `<referredEvaluation>` which links to an `<evaluation>` or another `<metaEvaluation>`. This allows one to reuse definitions (so the natural hydrogen `<metaEvaluation>` can be used in the assembly if many different TSL+fast `<metaEvaluation>`s). `<referredEvaluation>` has the following additional attributes:
  19.3.1 `stoichiometricFraction` tag lets one specify chemical or isotopic make-up if multiple `<referredEvaluation>`s are allowed
  19.3.2 `stoichiometricFraction` better add up to 1!
  19.3.3 `axisCoords` to specify where in the grid an evaluation sits.
19.4 Outside of parameter ranges in `grid`-like elements, the `<metaEvaluation>` does not exist
19.5 `<metaEvaluation>` only valid for listed projectile
19.6 Need tests to make sure every region in `<axes>` covered by a `<referredEvaluation>`.

Discussion point:

Is it possible to use say atomic weights instead of `stoichiometricFraction` to specify fractional composition of a material? This would simplify use in several transport code input decks.

Resolution:

Yes but at the cost of creating an unnecessary coupling between a `<metaEvaluation>` and the material database or there will be mistakes. Additionally, testing that the sum of fractional compositions sum to the correct value will be difficult.

Discussion point:

Is there a need for a separate `<metaEvaluation>`

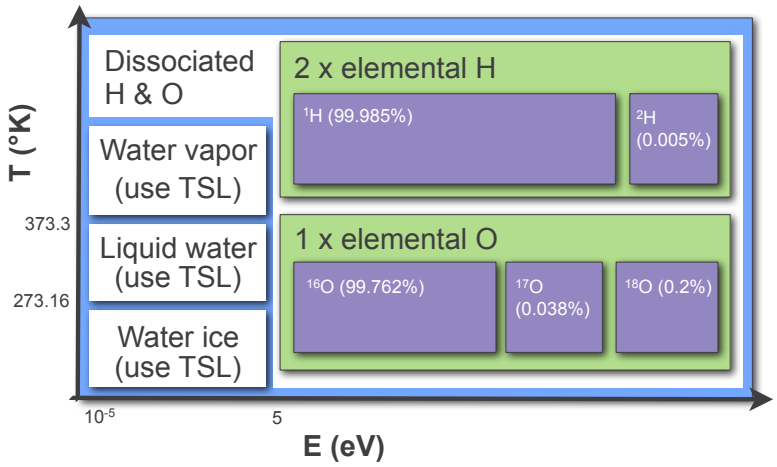


FIG. 13 A cartoon of a `<metaEvaluation>` that glues together thermal neutron scattering with the higher energy nuclear reaction data. Note that nesting `<metaEvaluation>`s can make the implementation of this quite simple.

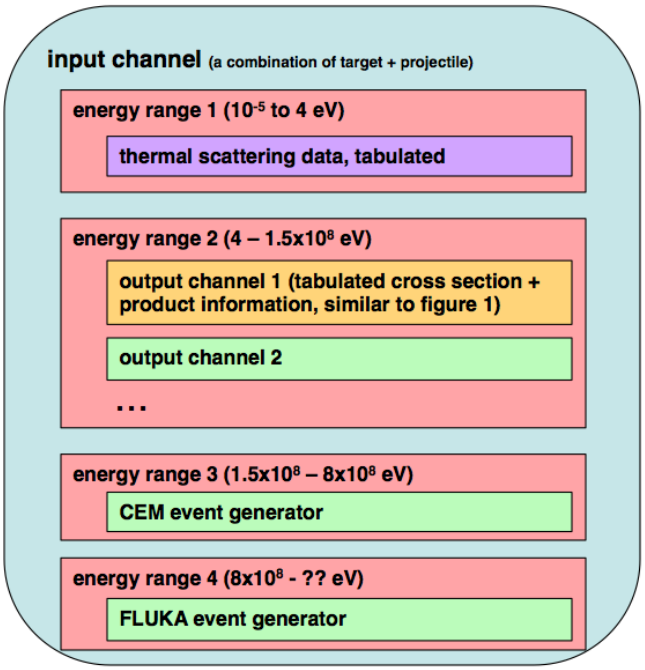


FIG. 14 Another cartoon of a `<metaEvaluation>` demonstrating its use to glue together different models from different energy regions. In this example, the event generators CEM and FLUKA are used at the highest energies.

concept to handle arbitrary projectiles so we needn't maintain several different (but nearly identical) element specifications?

Resolution:
Maybe allowing `any` or `*` as a projectile would serve this purpose. Alternatively, we could make the `projectile` attribute optional and if it is not present than the `<metaEvaluation>` is valid for all projectiles. Either way the links to the actual evaluation become meaningless. This requires some thought. Perhaps the resolution is to pre-make the elemental `<metaEvaluation>`s for the standard targets with fake URLs. Then users can swap-n-replace them with the correct URLs for their own needs. However, if one of the `<grid>`-like elements covered incident energy, there is a question of how to handle Q values and different channels opening up.

Discussion point:
We should consider specifying the interpolation scheme in each of the variables of the `<metaEvaluation>`'s `<axes>`.

Discussion point:
It was felt at the WPEC SG38 meeting in May 2014 that implementing `<metaEvalaution>`s amounts to "scope creep", meaning that this capability was not included in the original requirements. That said, it was generally agreed that this was a useful idea for data sharing.

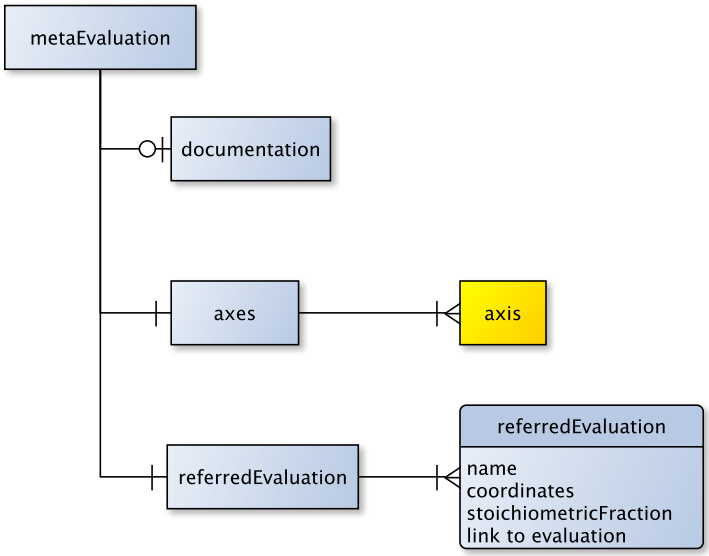


FIG. 15 The `<metaTarget>` element.

```
<library name="exampleLibrary">
  <documentation>...</documentation>
  <evaluations>
    <metaEvaluation name="water" projectile="n">...</metaEvaluation>
    <metaEvaluation name="Onat" projectile="n">...</metaEvaluation>
    <metaEvaluation name="Hnat" projectile="n">...</metaEvaluation>
    ...
    <evaluationLink name="HinH2O" projectile="n">...</evaluationLink>
    <evaluationLink name="OinH2O" projectile="n">...</evaluationLink>
    <evaluationLink name="H1" projectile="n">...</evaluationLink>
    <evaluationLink name="H2" projectile="n">...</evaluationLink>
    <evaluationLink name="H3" projectile="n">...</evaluationLink>
    <evaluationLink name="O16" projectile="n">...</evaluationLink>
    <evaluationLink name="O17" projectile="n">...</evaluationLink>
    <evaluationLink name="O18" projectile="n">...</evaluationLink>
    ...
  </evaluations>
  <covariances>...</covariances>
  <particles>...</particles>
</library>
```

FIG. 16 An example `<library>`. Here, the `<evaluationLink>`s connect to evaluations given in their own files.

Resolution:
The implementation of `<metaEvaluation>` specifications should be deferred until the main infrastructure is "in place" and we can devote time to creating the specifications and creating translators back and forth between LANL's `xsdir` format and other related formats at other data centers.

C. Reactions and reaction lists

Below the `<evaluation>` element, most of the data are stored in the `<reactions>` and `<resonances>` branches. The `<reactions>` element is one of three lists of reactions we will need. These lists all have a common structure, pictured in Figure 19. Each of the lists of reactions has a different use:

`<reactions>`: The list of reactions needed for most applications, especially particle transport.

`<summedReactions>`: The list of reactions whose cross section is a sum of the cross sections in the main

```
<metaEvaluation name="water" projectile="n">
    <documentation>...</documentation>
    <axes>
        <grid index="0" label="temperature" unit="K" style="boundaries">
            <values length="4">0.0 273.16 373.16 1e9</values></grid>
        <grid index="1" label="energy_in" unit="eV"  style="boundaries">
            <values length="3">1e-5 5</values></grid>
    </axes>
    <referredEvaluations>
        <referredEvaluation index="0" name="water ice" gridCoords="0,0" stoichiometricFraction="1.0">
            <link xlink:href="../../../metaEvaluation[@name='HinIce']"/></referredEvaluation>
        <referredEvaluation index="1" name="liquid water" gridCoords="1,0" stoichiometricFraction="1.0">
            <link xlink:href="../../../metaEvaluation[@name='HinH2O']"/></referredEvaluation>
        <referredEvaluation index="2" name="water vapor" gridCoords="2,0" stoichiometricFraction="1.0">
            <link xlink:href="../../../metaEvaluation[@name='WaterFreeGas']"/></referredEvaluation>
    </referredEvaluations></metaEvaluation>
```

FIG. 17 A contrived sample `<metaEvaluation>` specification, in this case for water. This files requires another `<metaEvaluation>` to specify the composition of dissociated water into the elements hydrogen and oxygen for the `'WaterFreeGas'` `<referredEvaluation>`. This then require other `<metaEvaluation>`s to specify the isotopic composition of naturally occurring hydrogen and oxygen. In both liquid water and ice, hydrogen is the dominant scatterer. As this example has two `<grid>`s (temperature and incident energy), each `<referredEvaluation>` requires two indices in their `gridCoords` attribute.

```
<metaTarget name="Onat" projectile="n">
    <documentation>...</documentation>
    <axes>
        <grid index="1" label="energy_in" unit="eV"  style="boundaries">
            <values length="3">1e-5 1e9</values></grid>
    </axes>
    <referredTargets>
        <referredTarget index="0" name="O16" gridCoords="0" stoichiometricFraction="0.99757">
            <link xlink:href="../../../evaluation[@name='O16']"/></referredTarget>
        <referredTarget index="1" name="O17" gridCoords="0" stoichiometricFraction="0.00038">
            <link xlink:href="../../../evaluation[@name='O17']"/></referredTarget>
        <referredTarget index="2" name="O18" gridCoords="0" stoichiometricFraction="0.00205">
            <link xlink:href="../../../evaluation[@name='O18']"/></referredTarget>
    </referredTargets></metaTarget>
```

FIG. 18 A contrived sample `<metaEvaluation>` specification of natural oxygen. In this case, there is only one parameter in the `<axes>` so the `gridCoords` attribute only has one index. As there is only one energy group, all `<referredTarget>`s have the same `gridCoords`.

`<reactions>` branch.

`<productionReactions>`: The list of reactions comprised entirely of production data (especially production cross sections).

1. `<reactions>` and `<summedReactions>`

Here we describe the two main kinds of reactions: exclusive (`<reaction>`) and inclusive (`<summedReactions>`). To understand the arrangement of data here and in the `<reaction>` and `<summedReactions>` elements, it is useful to have a mental model for particle transport.

For a neutral particle (such as a neutron or a gamma) with energy E traversing a material, the particle's mean free path (λ_{mfp}) is defined as

$$\lambda_{\text{mfp}} = \frac{1}{\sum_x n_x \sigma_{x,\text{tot}}(E)}, \tag{1}$$

where the sum goes over each component x of the material, and $\sigma_{x,\text{tot}}$ and n_x are the total cross section and number density respectively of component x. The mean free path determines the transit distance and time between "hard" nuclear collisions. The total cross section for each component should be tabulated in a `<summedReactions>` element.

For charged particles, the total cross section does not exist because of the Coulomb singularity in, for example, the elastic scattering reaction. Coulomb scattering is a "soft collision" in that the Coulomb force always acts to gently nudge the projectile at all distances. To implement Coulomb scattering in practice one divides up

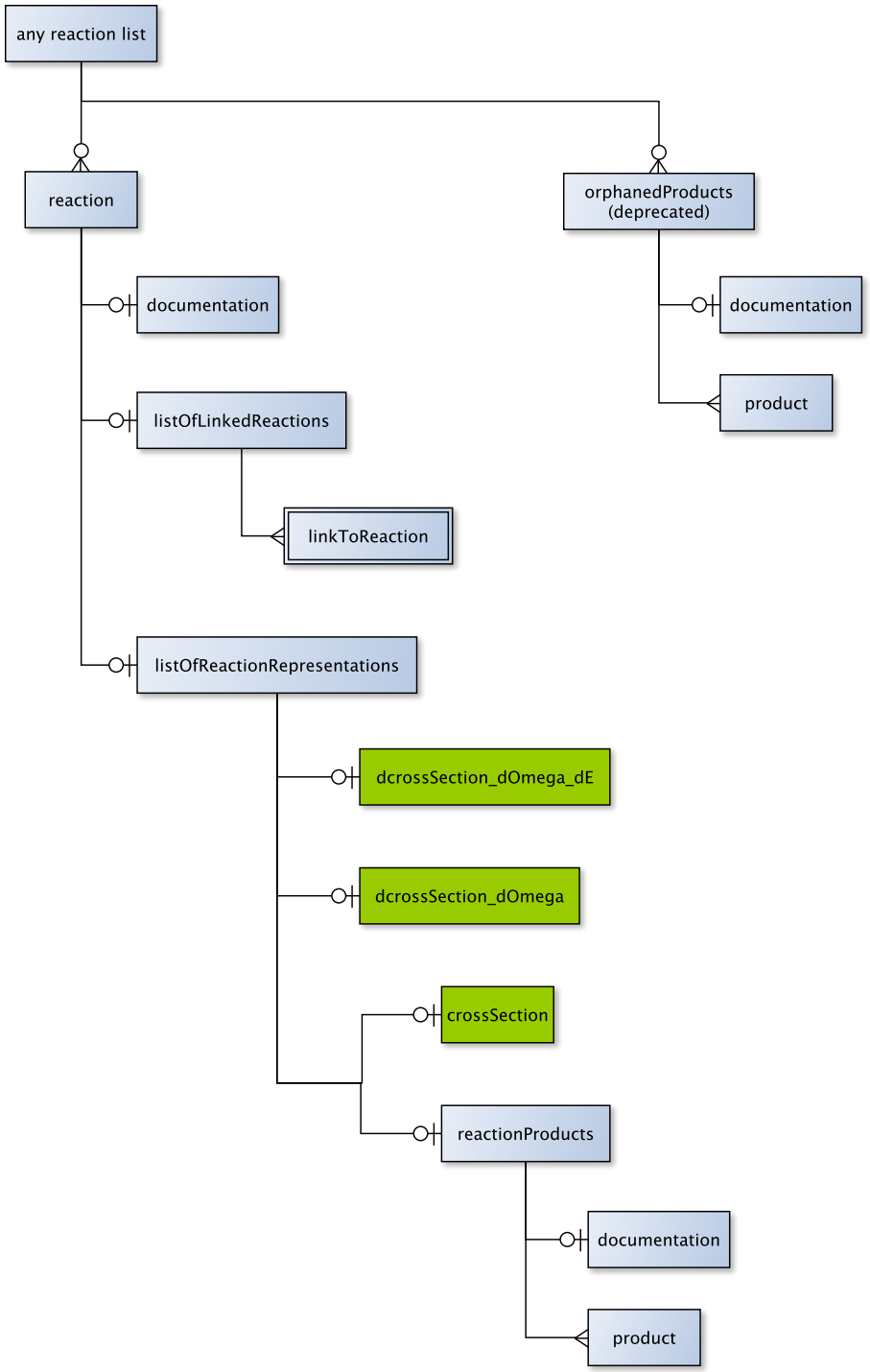


FIG. 19 A possible arrangement of inclusive and exclusive reactions in the `<reactions>` element. Note that there are three options for the double differential cross section data: a) store $d^2\sigma(E)/dE'd\Omega$; b) store $d\sigma(E)/d\Omega$ if the outgoing energy is fixed by kinematics; or c) store both $\sigma(E)$ and $P(E', \mu|E)$ separately. In some cases the total cross section $\sigma(E)$ is not defined and one must use one of the other options.

scattering events by angle relative to the center-of-mass momentum of the target and projectile. At small angles, one uses condensed history treatments (Seco, 2013). At large angles, one uses the large-angle Coulomb scattering plus "nuclear plus interference" approximation which treats the reaction as a "hard" inelastic collision. This mixture of treatments means the data cannot be heated.

For a "hard" collision, we can proceed as follows:

- Determine which material was hit by assigning probabilities proportional to $\sigma_x(E)$
- For the material, decide what reaction occurred by assigning probabilities proportional to the ratio of each partial cross section to the total cross section for that material
- Once the reaction is decided, determine what particles will be emitted
- Loop over emitted particles
  - If the multiplicity is not constant, sample the number of emitted particles using the energy-dependent multiplicity which may be given as a distribution (e.g., $P(\nu)$).
  - Depending on the kinematics (two-body or uncorrelated), sample the emitted particle's energy and/or angle
  - If particle or reaction product decays, follow the decays ...
- Where possible, use energy/momentum conservation to determine the recoil energy and momentum

So, it is clear we have a need for inclusive cross sections (especially the total cross section) and exclusive reaction data. The exclusive reaction data should go in the `<reaction>` elements in a `<reactions>` list. The requirements for a `<reactions>` list element are simple:

Requirement 20: **All reaction lists, especially** `<reactions>`

20.1 An optional `<documentation>` (not shown)
20.2 A list of `<reaction>`s with actual reaction data inside
20.3 Optionally, one or more `<unassociatedProduct>` elements storing multiplicities and distributions for products (usually gammas) that are observed but that the evaluator is unable to associate with a particular `<reaction>`.

Inclusive, or summed reactions, are slightly more complex and their requirements are:

Requirement 21: `<summedReactions>`

21.1 `<reaction>`s shall have common sense reaction designators (e.g., "total" or "absorption")
21.2 A (likely derived) cross section which includes list of links to the reactions whose cross section is meant to match the cross section tabulated here.
21.3 An optional `<documentation>` (not shown)
21.4 Optionally `<reactionProducts>` list with `<product>`s to store yields.
21.5 An optional (and deprecated) `<orphanedProducts>` branch. This allows for the storing of older evaluations. In many old ENDF evaluations, gammas were not associated directly with their source reaction, but were lumped in MT=3 or MT=4. This destroys energy balance and should be deprecated. This branch provides a spot for these unassociated gammas and other particles.

Discussion point:

As an alternative to Req. **21**.5, fake reactions could be added containing the orphaned products and symlinks could connect the `<reaction>` in the `<summedReactions>` to the `<reaction>` in the main `<reactions>` branch, allowing for a separation of transportable reactions from the sum rules.

In practice there are several inclusive reactions that we encounter in ENDF data:

1. Total cross section (a requirement for neutrons), allows evaluated version with covariance and the total may not exactly equal the sum of partial cross sections
2. ENDF sum rules (e.g., MT103 vs. MT600 – MT649)
3. Lumped cross sections for lumped covariances (should be deprecated!)

The total cross section is especially interesting. The total cross section is simple to measure and evaluators can both fit the data and model it with optical model calculations with high fidelity. The evaluation of the total cross section is usually quite solid with high quality covariances. However, there is no guarantee that if one sums up the partial cross sections of an evaluation that one will retrieve the total cross section. Therefore, while most evaluations will provide a total cross section (and place it in the `<summedReactions>` branch), a processing code will discard the evaluated total cross section and recompute the total from the sum of partials in order to ensure unitarity.

GND also encodes inclusive reactions in a `<sums>` element. This element includes a cross section (and this may be connected to covariance data). Additionally, there is a list of links to the reactions which are meant to be summed together to match the cross section data in the element. This element is used to implement all of the ENDF sum rules in Section 0.4.3.11 of the ENDF format manual (Trkov, 2009).

Discussion point:
The total cross section is often known to a higher precision experimentally than the sum of the cross sections that compose the total cross section. In this case, it should be acceptable to store the total cross section in a `<summedReaction>` element but denoted as an evaluated value. The other cross sections linked and listed as parts of the total can be summed and validated against the tabulated total cross section.

Resolution:
Agreed. This is implemented in GND-1.7.

SUGGESTED TEST: As `<summedReaction>`s have lists of links to the reactions they are derived from, we should check that the links are valid and, for example, `<crossSection>`s sum as promised.

Discussion point:
Should we also include weights in the list of links? This would add flexibility, but it is not clear how they would get used outside the case of lumped covariances. Incidentally, lumped covariances should be deprecated as they are not very useful.

Discussion point:
ENDF's 6 time-group beta delayed fission data should be stored in one or more "fission" `<reaction>`s. The ENDF time-groups are in principal derived from fission product yields and radioactive decay data. However, the time grouped data are often wrong for various reasons and useful 6 time-group data are actually fits to macroscopic spectra. While it might be tempting to link time-group data to fission product yield and decay data, it would make more sense to allow the time-group data to be denoted as the evaluated data.

2. `<productionReactions>` lists

As mentioned above, we also consider one other type of reaction list. Production reactions are common in derived, application-driven, data libraries and are used to store:

1. Gas production cross sections
2. Damage cross sections
3. Activation cross sections
4. Particle and isotope yields

A mis-filed production cross section is very dangerous. In the simplest case, a production cross section is a product of a particle's multiplicity and the cross section for that particle's production. For an (n,2n) reaction, this means a potential factor of two error. For gamma production, this could mean a factor of 100. Because the units on a production cross section and a regular cross section are the same, there may be no way to tell if an evauator mis-filed a production cross section as a normal cross section or vice-versa, leading to crazy energy balance bugs. However, this is avoidable if users store production data in a separate element with distinct naming. Therefore, we provide the `<productionReactions>` element to collect all of the production data together in a derived data file.

Discussion point:
All evaluated reactions are lumped together in `<reactions>` and `<summedReactions>` elements. Should we also group 'production' reactions (i.e., the sum of all reactions that produce product 'x')? Activation libraries could make heavy use of production cross sections.

Resolution:
The `activationFlag` should be used to denote an evaluation comprised of activation data. The production cross sections themselves are derived from cross sections in the `<reactions>` branch and stored in the `<productionReactions>` branch. See section VI for more detail.

3. `<reaction>` elements

With a reaction major arrangement, there is one common motif in the three different sub-libraries described in Section III, which for a lack of a better name, we call the `<reaction>` element. This element denotes one reaction that can be sampled in a Monte Carlo code. In it, we specify reaction `<crossSection>`s and the outgoing particle distributions for all emitted particles.

We comment that data computed from reconstructed resonances are placed in the appropriate sub-elements of a `<reaction>`. This reconstructed data would of course have hyperlinks connecting the derived data back to the data in the `<resonances>` branch.

Requirement 22: `<reaction>`

22.1 An optional `<documentation>`
22.2 The reaction data itself using at least one of these schemes:
  22.2.1 Option #1, breaking out cross sections

and outgoing distributions
22.2.1.1 A `<crossSection>`
22.2.1.2 A `<reactionProducts>` element listing the reaction `<product>`s. From the list of products it should be possible to reconstruct the reaction designator unless the reaction is denoted by some kind of alias.
22.2.2 Option #2, parameterized differential cross section: a `<dcrossSection_dOmega>` element (useful for two body reactions and Coulomb scattering)
22.2.3 Option #3, parameterized double differential cross section: a `<dcrossSection_dOmega_dE>` element (useful for N-body reactions and thermal scattering law data)
22.3 The ENDF MT if appropriate (deprecated)

Discussion point:

How does one implement breakup and/or multi-step reactions? The proper use of ENDF's LR flag scheme is complex. It is used for light element breakup, (n,gf) reactions and reactions which lead to unstable residuals (e.g., isomers) but whose half-lives are large enough that the residuals must be accounted.

Resolution:
This is handled by the `<decayProducts>` element in section III.E.1

Discussion point:

We can optionally store the Q value and threshold energies, but these are derivable if one knows the identity of the initial and final state particles. Requiring that they be given in a channel potentially introduces an internal consistency error if the values are not kept in sync with any external material property database.

Resolution:
Putting in the Q values in the `<reaction>` is useful, but we should not take the values seriously. Legacy ENDF evaluations specify the Q values for each reaction and these values may be chosen by the evaluator to get the correct threshold behavior of a reaction even if the Q value does not match that derivable from the incoming and outgoing particle masses. Furthermore, often the masses in ENDF are not stored with enough precision to obtain an accurate Q-value. So, if one intends to translate a file back into ENDF, one should specify the Q-value. The solution is possibly to derive the Q-value from masses by default and support legacy evaluations by including a deprecated option to specify the Q-value.

Discussion point:

Reconstructed cross sections and angular distributions could go here too

Resolution:
Agreed. This also provides a location in the hierarchy for other derived data used in processed data files.

4. Reaction designation

Common sense requires that we have a unique method for identifying a reaction that is preferably human readable and easily parsed by machine. Examples from GND (Mattoon, 2012) are shown in Table I.

We note that GND's scheme is more general than ENDF's MT designator and this scheme does not muddle MF and MT (as happens in the fission reactions in ENDF). In GND's scheme, the reaction designator is always unique and derivable from the reaction products (and their decay products if this is a breakup reaction). However, the user does have the ability to define their own reactions and add qualifiers to existing ones.

Discussion point:

The reaction designators in GND are little more than a "hash" in that they are unique computer generated shorthand based on the contents of a `<reaction>`'s `<reactionProducts>` list. Fudge does not use them, relying instead on a separate unique `id` attribute. That said, they are very useful for navigation of a file by humans.

GND reaction label		ENDF MT
n + Pu239	→ n + Pu239	2
n + Pu239	→ n + Pu239 [compound elastic]	
n + Pu239	→ n[multiplicity:'2'] + Pu238	16
n + Pu239	→ n[multiplicity:'3'] + Pu237_e1	
n + Pu239	→ n + Pu239_e1	51
n + Pu239_m1	→ n + Pu239_c	91
n + Pu239	→ Pu240 + gamma	102
n + Pu239	→ Pu240_e1 + gamma	
C12 + Pu239	→ C12_e2 + Pu239_e1	
n + Be7	→ (Be8 → He4[multiplicity:'2'])	

TABLE I Example of reaction labels in GND. ENDF MT numbers are listed when possible. Some GND reactions have no MT equivalent. From Ref. (Mattoon, 2012)

Whatever scheme we use, users should immediately be able to tell what the target, project, and all reaction products are. We should also be able to define aliases for commonly understood names like "fission" and "spallation". Additionally, we should be able to describe some reactions with an additional qualifier which could allow us to, for example, break "elastic" scattering into "compound elastic" and "shape elastic". This would allow us to better represent the reaction distributions since neutrons from "compound elastic" reactions are typically isotropic in the residual nucleus rest frame while "shape elastic" neutrons are typically forward–peaked.

In whatever scheme is finally agreed on for the reaction designators it should follow these recommendations:

Requirement 23: Reaction designation

- **23**.1 Should be shared/agreed upon with EXFOR
- **23**.2 Should not be limited to simple targets (we need to denote thermal neutron scattering data)
- **23**.3 Support aliases for things like "`elastic`", "`total_fission`", "`capture`" and "`spallation`".
- **23**.4 Support need to distinguish input vs. output channels
- **23**.5 Allow uncorrelated particle emission
- **23**.6 Support processes with non-constant multiplicities
- **23**.7 Support sequential processes (esp. 2-body)
- **23**.8 Support qualifiers such as "`compound_elastic`" and "`shape_elastic`" which allow evaluators to split up reactions with common final products but different reaction mechanisms and/or kinematics.

SUGGESTED TEST: Test that all reaction designators are unique since they are keys and we must avoid key collisions.

D. Cross sections

1. Integrated cross sections: $\sigma(E)$

A `<crossSection>` element is needed to store ENDF's MF=3 or 23 data. It would be used for both atomic scattering and nuclear reaction data.

Requirement 24: `<crossSection>`

- **24**.1 A `<crossSection>` element is either:
  - **24**.1.1 At least one version of the data containing an `<interp1d>` element with a dependent variable (the cross section itself) given in units of area and independent variable (projectile's incident energy) in units of energy. The first energy point could lie at the (real or effective) threshold or at the lowest energy supported by the encapsulating evaluation. A `<crossSection>` is assumed to be zero outside of the specified energy region
  - **24**.1.2 a link to the resonance region that one must reconstruct in order to retrieve the cross section data tables
- **24**.2 These are mid-level container elements (see requirements **11**).
- **24**.3 A `<crossSection>` element may have `<documentation>`.
- **24**.4 An optional unresolved resonance cross section probability table (see section VI)

Discussion point:

Why not leave the cross section "undefined" outside of the specified range? Then one can leave it up to the user/code to decide what the cross section should be (zero or non-zero constant value or whatever).

Resolution:

By not defining it in the specifications, we allow for possibly different behaviors in different user codes. Specifying the behavior outside the defined range ensures consistency between user code implementations.

Discussion point:

It was suggested to give cross sections as ratios, for example ratios to total? This was seen as a way to eliminate sum rule failings.

Resolution:
However, this would intentionally introduce synchronization troubles and require rewriting a lot of code to take advantage of. In addition, we have substituted one sum rule (summing to a cross section) with another (summing to 1), so we really haven't gained anything.

2. Differential cross sections: $d\sigma(E)/d\Omega$ and $d^2\sigma(E)/d\Omega dE'$

There are many cases where it is more convenient to write two-body scattering data as $d\sigma(E)/d\Omega$ rather than as a separate cross section $\sigma(E)$ and angular distribution $P(\mu|E)$ where $d\sigma(E)/d\Omega = \sigma(E)P(\mu|E)$. Cases include:

- Thermal Scattering Law (TSL) elastic data, see Section V.G
- Charged particle scattering data, see Section V.B
- Photo-atomic data described with the Klein-Nishina (KN) formula, see Section V.A.1

Indeed, in the case of charged particle scattering data, the singularities in the Rutherford cross section prevent us from integrating to find the total cross section $\sigma(E)$. Therefore, we must provide a facility for flagging a reaction as a special parameterized two-body reaction and a facility for storing $d\sigma(E)/d\Omega$.

Requirement 25: `<dcrossSection_dOmega>`

25.1 The actual implementation (which depends on the nature of the described data).
25.2 An optional `<documentation>`
25.3 A flag to denote whether to use relativistic or non-relativistic kinematics when handling this channel
25.4 These are mid-level container elements (see requirements **11**).

In the case of Thermal Scattering Law (TSL) inelastic data (see Section V.G), it is more convenient to write two-body scattering data as $d^2\sigma(E)/d\Omega dE'$ rather than as a separate cross section $\sigma(E)$ and energy-angle distribution $P(E', \mu|E)$ where $d^2\sigma(E)/d\Omega dE' = \sigma(E)P(E', \mu|E)$. Therefore, we must provide a facility for flagging a reaction as a special parameterized two-body reaction and a facility for storing $d^2\sigma(E)/d\Omega dE'$.

Requirement 26: `<dcrossSection_dOmega_dE>`

26.1 The actual implementation (which depends on the nature of the described data).
26.2 An optional `<documentation>`
26.3 A flag to denote whether to use relativistic or non-relativistic kinematics when handling this channel
26.4 These are mid-level container elements (see requirements **11**).

E. Products and product lists

1. Product list elements

Both particle decays and induced reactions produce reaction products. In addition, in legacy ENDF there is the possibility of "orphaned products", that is, products that are not correctly associated with their parent reaction. This is uncommonly seen in ENDF MT's 3 and 4 outgoing photon data. Therefore we will need an element that encapsulates lists of reaction products. To support ENDF, we will require three different kinds of product lists:

- A `<reactionProducts>` element to list the reaction `<product>`s of an induced reaction. In GND-1.7, a `<reactionProducts>` element is referred to as `<outputChannel>`.
- A `<decayProducts>` element to list the daughter particles into which a parent `<product>` decays. This is for use in particle properties databases, but uses the same arrangement of data as other product list elements.
- An `<orphanedProducts>` element to list `<products>` that are not associated with any reaction, as is found in older evaluations (especially for gamma products).

Although each of these product lists share nearly the same structure, we assume that it is easier to work with unique names as it helps to denote the context in which they are used. Whether they have separate names or the same name is up to the authors of the specification document. Each list of reaction products has additional individual requirements that we list here. Their use in either the context of induced reactions or the decay of particles are discussed later in this document.

In Fig. 20 we show the common structure of all of these lists. From the list of products it should be possible to reconstruct the reaction designator in the `<reaction>` element (see section III.C.3).

Requirement 27: `<reactionProducts>`

27.1 List of `<product>` elements
27.2 The `kinematicType` (e.g., two-body, uncorrelated). Elastic reactions and all resonance reactions using the R matrix formalism are two-

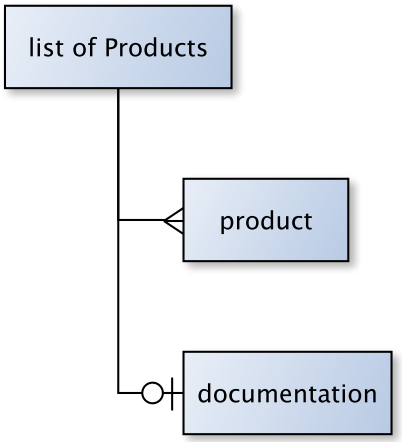


FIG. 20 Common arrangement of a product list element such as `<reactionProducts>`, `<decayProducts>` or `<orphanedProducts>`.

body reactions. GND refers to these by the name `<genre>`. Table II lists allowed kinematic types.

27.3 A flag to denote whether to use relativistic or non-relativistic kinematics when handling this channel

27.4 Optionally the Q value of the reaction for backwards compatibility with the legacy ENDF format

Requirement 28: `<orphanedProducts>`

28.1 List of `<product>` elements

28.2 The `kinematicType` (most likely uncorrelated). Table II lists allowed kinematic types.

28.3 A flag to denote whether to use relativistic or non-relativistic kinematics when handling this channel

28.4 Optionally the Q value of the reaction for backwards compatibility with the legacy ENDF format

Requirement 29: `<decayProducts>`

29.1 Intended mainly for use in particle property databases

29.2 List of links to decay products or the decay `<product>`s themselves

29.3 The `kinematicType`. Table II lists allowed kinematic types.

29.4 `Q`, For new evaluations, this is optional but should sync with material properties. For legacy evaluations it must be retained.

29.5 `lifetime`, a material property but it may be useful to repeat it here, so it is optional

TABLE II Kinematic types for an attribute for `<reactionProducts>`, `<decayProducts>` and `<orphanedProducts>` elements.

`kinematicType`	Description
two-body	only two products are emitted per step in a reaction, the products are correlated, and only the center-of-mass angular distribution is needed in order to calculate the double-differential distribution
uncorrelated	the products are treated as being uncorrelated from each other, and a complete double-differential distribution is required for each product
correlated	proposed
fission	proposed

Discussion point:

At some point we may have to develop a treatment for correlated particle emission. At the very least, there could be a new `<correlatedProducts>` list which denotes what particles are correlated and perhaps giving conditional probability tables for (E', μ) of each emitted particle or perhaps a list of pre-generated correlated events. Another option is to provide the inputs for an external event generator.

2. `<product>` elements

In the legacy ENDF format, outgoing product data are stored in a variety of ways, potentially leading to confusion for the user:

1. Outgoing neutron energy distributions are stored in an MF=5 file and angular distributions in an MF=4 file, assuming the neutron energy and angular distributions are uncorrelated. Alternatively, both may be stored in an MF=6 file. Correlated double differential neutron data can only be stored in an MF=6 file. Neutron multiplicities from fission are stored in MF=1, MT=452, 456, and 455 files.

2. Outgoing charged particle data are stored exclusively in MF=6 files.

3. Outgoing gamma data can be stored in MF=6 files or in a combination of MF=12 (for multiplicities and discrete level energies), MF=13 (production cross sections), MF=14 (angular distributions) and

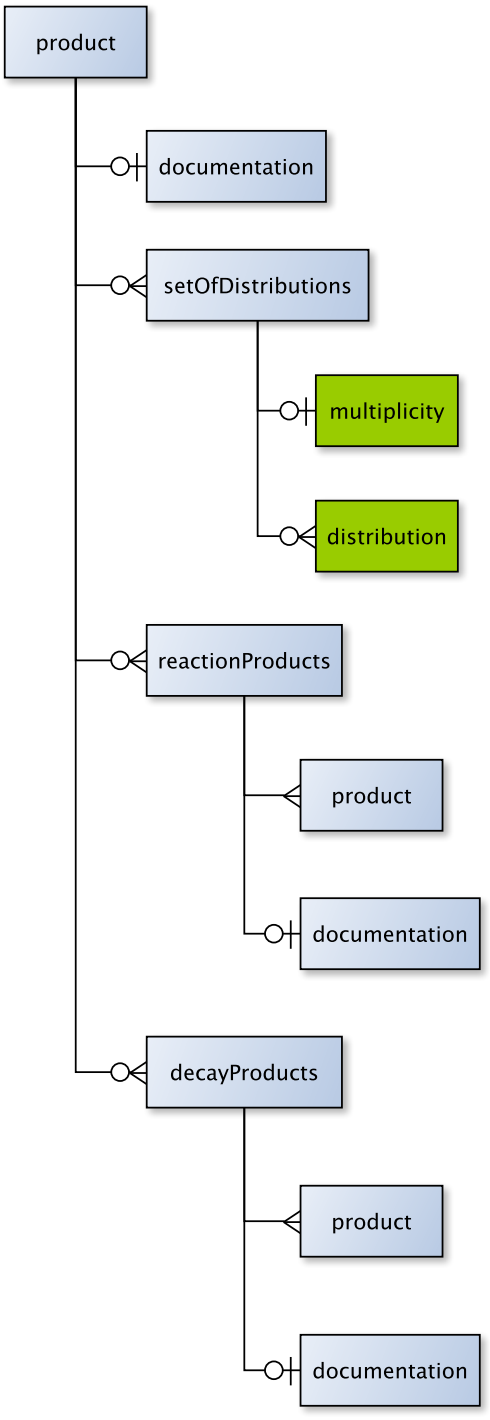


FIG. 21 Overview of a `<product>` element.

MF=15 (energy distributions). Additionally, delayed gamma data from fission are stored in MF=1, MT=460.

4. Additionally, the energy released from fission is stored in MF=1, MT=458 and is not associated with the produced particle.

We would like to simplify and unify these options into a simple product (and daughter) elements.

Each induced reaction or spontaneous decay yields products that are grouped in one of the list elements defined in the previous section. The `product` element must support storing (at least) a multiplicity and outgoing distributions.

The `<product>` structure is illustrated in Figure 21. We note that, in this figure, the `<product>` element has both documentation as well as all of the derived and evaluated distributions and multiplicities corresponding to the reaction or decay product.

Often these products further react either because they are an intermediate state (as in a breakup reaction), or metastable or unstable (as in decay data). To enable this, we allow the `<product>` element to have both `<reactionProducts>` and `<decayProducts>` within. This enables multistep, breakup or decay reactions in an (hopefully) obvious way. Strictly speaking, if one has an external (or even internal to the evaluation) particle properties database, the `<decayProducts>` element could be placed with the particle properties rather than in a `<reaction>` element.

Each `<product>` should have:

Requirement 30: `<product>`

30.1 The particle's identity. For a transportable particle this refers directly to the particle. For a particle (nucleus or atom) with excitation levels, this refers to both the particle and the excitation level.

30.2 The multiplicity

30.3 An optional ENDF conversion flag (deprecated) to tell the user whether the original ENDF product data was in MF=6 or a combination of MF=4,5 or MF=12,13,14,15 data to aid in reverse translation.

30.4 All specified outgoing particle distributions for that particle. These are mid-level container elements, see requirements **11**. This can include things not pictured in figure 21 such as the mean energy and forward momentum of the particle to enable heating calculations, or a transfer matrix for group-wise deterministic calculations

30.5 If the product can subsequently decay, then a `<decayProducts>` element should be allowed.

30.6 If the product is an intermediate state in a breakup or multi-step reaction, then a `<reactionProducts>` element is allowed.

Discussion point:

The nesting of product lists inside a `<product>` list leads to schema recursion. This is allowed in XML

and other hierarchical formats, but may lead to complicated schema coding and should be avoided if possible.

Resolution:
Noted. One solution is to provide multiple product list element tags with names that depend on the nesting depth.

F. Distribution and distribution lists

1. <distributions> and <distribution> elements

The <distributions> element contains all of the outgoing probability tables associated with a reaction product. Each probability table is contained inside a <distribution> element corresponding to some variation of $P(\mu, E'|E)$ that can be used for a transport application. More than one distribution may be stored inside the <distributions> (i.e., both original and derived distributions may be stored as separate <distribution> elements).

Several different types of distribution data are possible. These include angular distributions $P(\mu|E)$, used for two-body reactions where knowledge of the outgoing angle of one product is sufficient to reconstruct all reaction kinematics for both products[4], and double-differential distributions $P(\mu, E'|E)$, used to give the full probability density for products coming from more complex reactions. Many evaluations use a special case of $P(\mu, E'|E)$ by making the simplifying assumption that the outgoing μ and E' distributions are <uncorrelated> and can be stored separately as $P(\mu|E) \cdot P(E'|E)$ (e.g., MF 4 and MF 5 in ENDF-6 rather than in MF 6).

In GND the <distributions> element contains one or more *forms* of the distribution, each of which stores one representation of the distribution. For example, for a two-body reaction an evaluator can use a Legendre form (i.e., representation) for the angular data for one of the products. In addition, a processing code can generate a pointwise representation of the Legendre data and this can also be stored in the <distributions> element in a pointwise form. All distribution forms except for the <uncorrelated> form contain one dataset (i.e., low level functional container). The <uncorrelated> component contains two datasets: one for the angular data and one for the energy data.

The organization used in GND has some advantages: it starts by storing the *type* of distribution (at the component level) and then the details of *how* that distribution is being stored (at the form level). However, this two-level organization has drawn some criticism, partly due to confusion over the special treatment required for <uncorrelated> distribution components, and also partly due to its limited capability for handling different forms of distribution in different energy regions (like ENDF, GND permits mixing Legendre distributions at low incident energy with pointwise angular distributions at higher incident energy, but other combinations are not supported).

Unfortunately, in the opinion of the GND authors there is no way to avoid treating <uncorrelated> as a special case since it must contain both $P(\mu|E)$ and $P(E'|E)$; neither by itself is sufficient to describe the distribution.

Discussion point:
Some other possible data hierarchies should be considered, however. For example, instead of nesting forms inside of component elements, both the type and the form of distributions could be given as metadata, as in <distribution type="angular" form="pointwise">. Also, a general-purpose <piecewise> distribution form could be defined, containing two or more distribution <region> elements each of which applies over a specified range of incident (and / or outgoing) energies.

Requirement 31: <distributions>

31.1 List of <distribution>s or links to them
31.2 All of the energy, angle and energy-angle PDFs used in the ENDF-6 format in Chapters 4, 5, 6, 12, 13 and 14 of (Trkov, 2009) and in the ENDL format (Howerton, 1983) shall be supported.
31.3 These are mid-level container elements (see requirements **11**).

Discussion point:
It was requested that we allow a <distribution> to link to another <distribution>. This construct would be helpful in storing processed data at various temperatures for Monte Carlo transport where one only heats the cross sections. One could then generate the heated cross sections and store the cross sections in evaluations at different temperatures and connect them with the metaEvaluation markup. To reduce the massive redundancy in the outgoing distributions (they never get heated), all the distributions in the heated evaluations could then link back to the zero temperature file's distributions. In fact, it may be more economical to have

[4] For a two-body reaction, the double differential distribution is related to the angular probability distribution by $P(\mu, E'|E) = \delta(E' - E'(E, \mu))P(\mu|E)$.

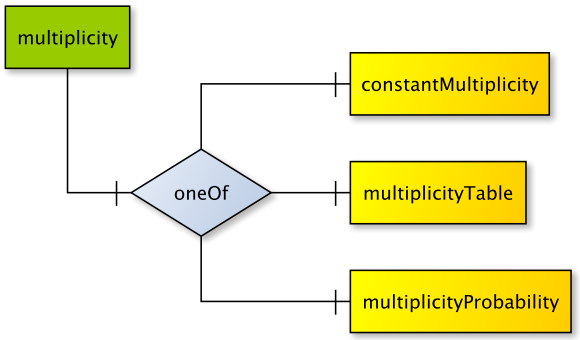


FIG. 22 Overview of a `<multplicity>` element.

the entire `<reactionProducts>` element link to another.

Resolution:
This discussion needs to be revisited and updated: the `<representations>` section now supports storing multiple temperatures inside a single `<evaluation>`. A representation does not need to overwrite all types of data: in this example, a heated representation of data could overwrite a reaction cross section but not the corresponding distributions.

2. Multiplicities

A `<multiplicity>` element would be used by atomic scattering and nuclear reaction data. It is analogous to ENDF's MF=12 (for gammas) or MF=1, MT's 452, 455 or 456 (fission $\bar{\nu}$'s), but simplifies and unifies the structure by defining one common scheme for multiplicity specification.

Requirement 32: `<multiplicity>`

32.1 A non-constant `<multiplicity>` element consists of at least one representation containing an `<interp1d>` with a dependent variable (the multiplicity itself) given in units of number of emitted particles and an independent variable (projectile's incident energy) in units of energy. The first energy point could be (real or effective) threshold or the lowest energy supported by the encapsulating evaluation. The `<multiplicity>` is assumed to be zero outside of the specified energy region. If the `<multiplicity>` is variable and given as a non-integer (as is common for fission $\bar{\nu}$), it is up to the code using the data to interpret the data correctly

32.2 Shall support $P(\nu|E)$ for fission neutrons.

32.3 These are mid-level container elements (see requirements **11**).

Discussion point:
Early versions of GND allowed storing constant multiplicities as attributes rather than a elements. For consistency, should we require that all multiplicities be given in their own `<multiplicity>` element inside a `<product>` element and eliminate the idea of storing constant multiplicities as an attribute? This would make for simpler coding of an API and clearer data files at a small cost of verbosity.

Resolution:
Agreed

Discussion point:
Should we allow multiple names for the same element? Physically, `<promptNubar>` is just the average multiplicity for the prompt neutron product, so it could be stored just like any other product multiplicity. However, `<promptNubar>` is easier to search for...

Resolution:
No, it leads to a more complex data structure.

Discussion point:
(Actually a continuation of the previous discussion point): in GND/XML, converting from metadata to a unique element only takes up a little more space. However, when translating to HDF5 each element is converted into a unique 'group' which takes up a minimum of about 1.3 kB, so the difference between metadata and element can become significant.

Resolution:
This is a serious issue with one particular potential implementation (HDF5). Therefore it will be up to the developers of the API for the data to deal effectively with this issue.

G. Resonances

The resonance region is a surprisingly complicated part of any evaluation. A resonance is a sharp feature in

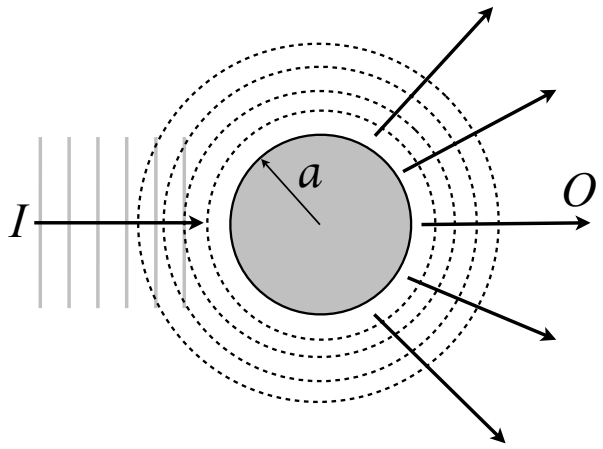


FIG. 23 A cartoon representation of R matrix theory. We first divide the universe into inside a box and out. Inside the box is the reaction zone, that we have little chance of modeling correctly without a lot of work. Outside the box we write all incoming and outgoing relative two-body scattering states in a basis of analytic wave functions, usually taken to be free ones. We then match wave functions on the box boundary.

a cross section caused by a target capturing a projectile and forming a compound nucleus with a specific J^{Π} and energy. The compound nucleus later decays by particle emission. For charged particle incident reactions, there are typically very few resonances in the observed cross sections because the resonances are suppressed by Coulomb effects. On the other hand, in a neutron induced reaction there can be thousands of observed resonances.

To understand the hierarchy of resonance data, it is helpful to understand a little about R matrix theory. In it, the universe is divided into the inside and outside of a spherical box. Inside the box is the reaction zone, where all the interesting nuclear (or other) reaction business occurs (see Figure 23). We have little chance of modeling what goes on in the box correctly without a lot of work. Outside the box we write all incoming and outgoing relative two-body scattering states in a basis of analytic wave functions, usually taken to be free ones. We then match wave functions on the box boundary. This matching is done in a clever way involving Bloch surface operators on the box boundary and from this we arrive at a Green's function of the projected Bloch-Schrödinger equation, also known as the R matrix. The elements of the R matrix are:

$$R_{cc'} = \sum_{\lambda} \frac{\gamma_{\lambda c}\gamma_{\lambda c'}}{E_\lambda - E}, \tag{2}$$

The factor $\gamma_{\lambda c}$'s are the reduced widths for channel c, E_λ becomes the resonance energy (it is a pole in the Laurent series expansion of the Green's function) and λ is the resonance (pole) index. The channel index c contains all the quantum numbers needed to describe the two-particle state and all of those quantum numbers are described in the `<channel>` and `<spinGroup>` element markups below. The channel index may refer to an incoming or an outgoing channel.

With the R matrix, it is possible to compute exactly the channel-channel scattering matrix $U_{cc'}$:

$$\begin{aligned} U_{cc'} =& e^{-i(\varphi_c+\varphi_{c'})}\sqrt{P_c}\sqrt{P_{c'}} \\ & \times \{[\mathbf{1}-\mathbf{R}(\mathbf{L}-\mathbf{B})]^{-1}[\mathbf{1}-\mathbf{R}(\mathbf{L}^*-\mathbf{B})]\}_{cc'} \end{aligned} \tag{3}$$

where $\mathbf{L}$ is the matrix of the logarithmic derivatives of an outgoing channel function with diagonal elements

$$L_c \equiv a_c \frac{O_c'(a_c)}{O_c(a_c)} = \left[r_c \frac{\partial \ln O_c}{\partial r_c} \right]_{r_c=a_c}. \tag{4}$$

We write

$$L_c = S_c + iP_c. \tag{5}$$

The penetration factor is $P_c = \Re L_c$ and the shift factor is $S_c = \Im L_c$. Both take their names from their function in the simple complex square well scattering model. φ_c is the phase factor

$$\varphi_c \equiv \arg O_c(a_c) = \arctan \frac{\Im O_c(a_c)}{\Re O_c(a_c)} \tag{6}$$

The constant B_c is the so-called "boundary parameter" which must be specified to correctly compute the scattering matrix, but is not always clearly given. We note that the non-trivial nature of even the hard-sphere penetrability, shift and phase give rise to the potential scattering behavior of the cross sections even when there are no resonances.

With the scattering matrix, one can compute all channel cross sections, the total cross section, and all angular distributions. The angle integrated cross section can be written as sum over all entrance channels $c = \{\alpha J \ell s\}$ and exit channels $c' = \{\alpha' J' \ell' s'\}$ that lead from partition α to α':

$$\sigma_{cc'} = \pi \lambda\!\!\!^{-2}_c g_c |\delta_{cc'} - U_{cc'}|^2 \tag{7}$$

The total cross section for channel c is

$$\sigma_c \equiv \sum_{c'} \sigma_{cc'} = 2\pi \lambda\!\!\!^{-2}_c (1 - \Re U_{cc}) \tag{8}$$

The factor of g_c is the probability of getting the correct J from the spins of the collision partners (according to Fröhner) and is $g_c = (2J+1)/((2i+1)(2I+1))$.

The Blatt-Biedenharn equation (Blatt, 1958) is used to construct the $d\sigma_c/d\Omega$ for two body channels in the center-of-momentum. In the ENDF formatted libraries, the Blatt-Biedenharn equation is usually used for elastic channels. Although $d\sigma_c/d\Omega$ can be written as a Lorenz covariant quantity, we will write the outgoing dependence on angle in their pair center of mass frame and the incident energy in the laboratory frame.

For non-identical particles with arbitrary spin, we have

$$\frac{d\sigma_{\alpha,\alpha'}(E)}{d\Omega} = \frac{1}{k^2(2i+1)(2I+1)} \sum_{s,s'} \sum_{L=0}^{\infty} B_L(\alpha s, \alpha' s'; E) P_L(\mu) \tag{9}$$

and

$$\begin{aligned} B_L(\alpha s, \alpha' s'; E) =& \frac{(-)^{s-s'}}{4} \sum_{J_1,J_2} \sum_{\ell_1,\ell_2} \sum_{\ell_1',\ell_2'} \bar{Z}(\ell_1 J_1 \ell_2 J_2 s L) \bar{Z}(\ell_1' J_1 \ell_2' J_2 s' L) \\ & \times (\delta_{\alpha\alpha'}\delta_{\ell_1\ell_1'}\delta_{ss'} - U^{J_1}_{\alpha\ell_1 s,\alpha'\ell_1' s'}(E))^* (\delta_{\alpha\alpha'}\delta_{\ell_2\ell_2'}\delta_{ss'} - U^{J_2}_{\alpha\ell_2 s,\alpha'\ell_2' s'}(E)) \qquad (10) \\ =& \frac{(-)^{s-s'}}{4} \sum_{c_1=\{\alpha\ell_1 s_1 J_1\}} \sum_{c_1'=\{\alpha'\ell_1' s_1' J_1'\}} \sum_{c_2=\{\alpha\ell_2 s_2 J_2\}} \sum_{c_2'=\{\alpha'\ell_2' s_2' J_2'\}} \bar{Z}(\ell_1 J_1 \ell_2 J_2 s L) \bar{Z}(\ell_1' J_1 \ell_2' J_2 s' L) \\ & \times \delta_{ss_1}\delta_{s's_1'}\delta_{J_1J_1'}\delta_{ss_2}\delta_{s's_2'}\delta_{J_2J_2'} (\delta_{c_1c_1'} - U_{c_1c_1'}(E))^* (\delta_{c_2c_2'} - U_{c_2c_2'}(E)) \qquad (11) \end{aligned}$$

where

$$\bar{Z}(\ell_1 J_1 \ell_2 J_2, s L) = \sqrt{(2\ell_1+1)(2\ell_2+1)(2J_1+1)(2J_2+1)} (\ell_1 \ell_2 00, L0) W(\ell_1 J_1 \ell_2 J_2, s L) \tag{12}$$

and $W(\ell_1 J_1 \ell_2 J_2, sL)$ is a Racah coefficient.

We use the notation $\sum_c = \{\alpha \ell s J\} = \sum_\ell \sum_s \sum_J$. The ENDF manual uses the notation $\sum_c = \sum_\ell \sum_s$, so it needs an extra sum over J_1 and J_2 (Trkov, 2009). We note that for charged particle incident reactions these equations are modified in a well known manner described in (Thompson, 2009).

This is detailed in several places including (Blatt, 1958; Descouvemont, 2010; Froehner, 2000; Lane, 1958). Given the mathematical completeness of the theory, it is no surprise that we mostly just view the R matrix parameters as simple fit parameters and then essentially get all of this for free.

In ENDF and in our proposed data structure, the parameters defining resonances are stored, rather than the reconstructed cross sections. This is done partly for a more compact presentation of the data and partly for a more exact representation of the data. In this section, we describe resonance data as in ENDF's MF=2, MT=151. Our proposed hierarchy is given in Figure 24. These data describe resonances that are observable in neutron cross sections for $E = 0$ eV $\rightarrow$ 100's keV (or higher for charged particles). In ENDF, these data are only used for neutrons, but should be legal for charged particle reactions and even photonuclear data.

Finally, we comment that the R matrix approach works for *any* two-body reaction, relativistic or not, as long as the incoming and outgoing relative states can be clearly defined. In the nuclear data community we often forget this fact. As a result, the ENDF format never really had the ability to represent charged particle data in an R matrix inspired form, reducing the quality and scope of data available to several communities who need it:

- Inertial Confinement Fusion community needs all sorts of charged particle incident data
- Astrophysical community needs the (p, γ) reaction among many others
- Ion beam analysis needs charged particle incident data
- For Nuclear Resonance Fluorescence, need to support (γ, γ') data
- FRIB Support: all of the FRIB experiments will be with beams of rare isotopes. Since a target of neutrons cannot be made, nearly all of the beam-target experiments will be charged-particle reactions, probably on hydrogen, deuterium and helium targets. There have been some advances in the application of R-matrix methods to these reactions.

Discussion point:

By allowing the possibility of creating an evaluation in which one specifies the URR without an RRR, we allow the possibility of an evaluation in which we do not know any resonances, but only the average parameters (perhaps via systematics). This was attempted in the original ENDF ^{240}Am evaluation which has since been superseded.

Discussion point:

The careful reader might notice that the Adler-Adler resolved resonance format is not mentioned anywhere else in this document. This format was deprecated before the release of ENDF/B-VII.0 and is not

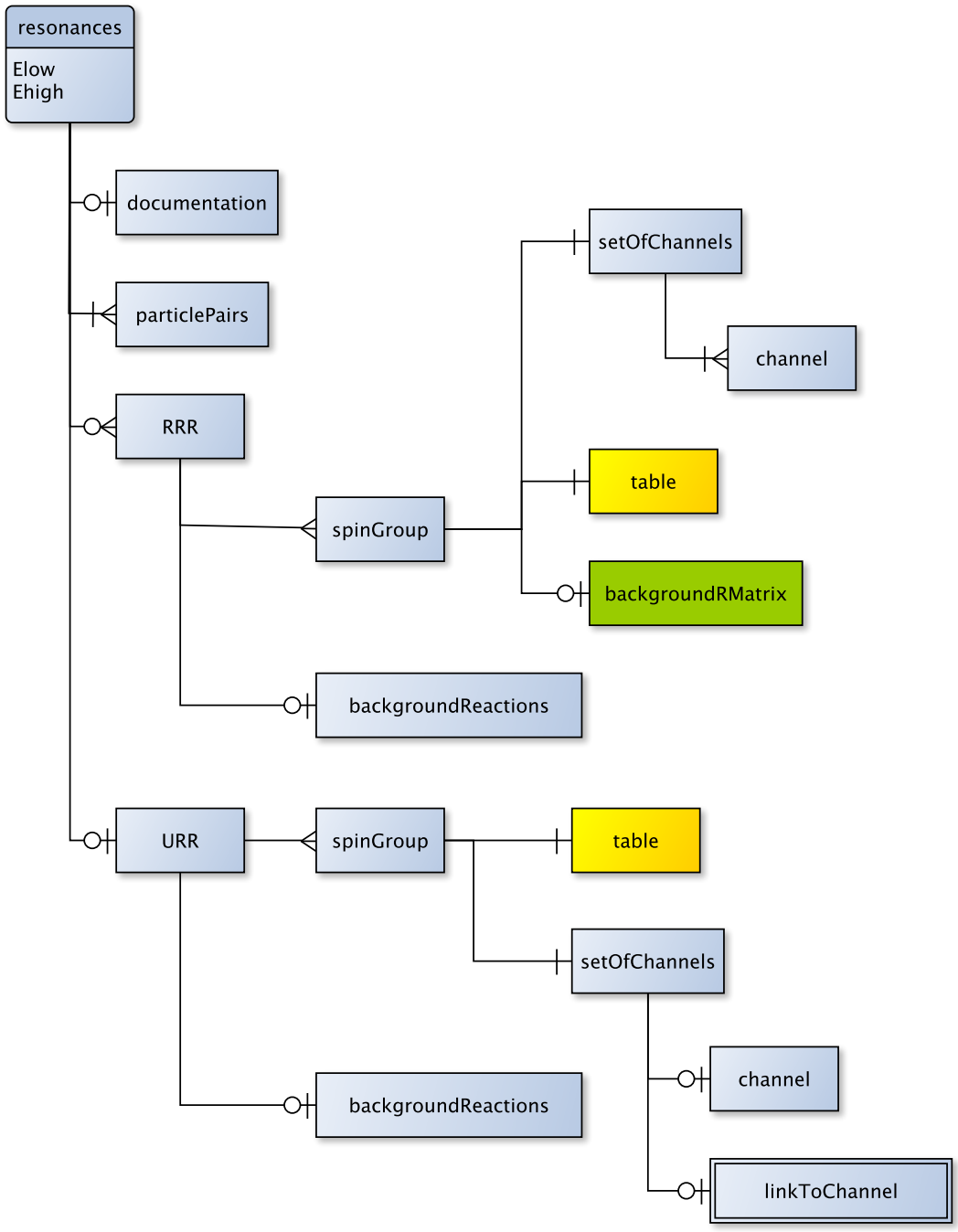


FIG. 24 Our proposed resonance data hierarchy. Note the separate background correction possibilities in the RRR and URR regions.

used in any modern data libraries. If data originally in the representation are needed in the future, one will need to covert the Adler-Adler resonances into a supported format.

Requirement 33: `<resonances>` **element**

33.1 Optional documentation
33.2 A list of the channels referred to in this evaluation. Traditional ENDF SLBW, MLBW and Reich-Moore formats support only capture, elastic, fission, total and a catch-all competitive channel. The R matrix formalism can support *any* two-body final state.
33.3 Optionally one or more resolved resonance region (RRR), although multiple RRR is deprecated
33.4 Optionally an unresolved resonance region (URR)
33.5 Upper and lower energy limits delineating the range of applicability of the resonance region.

1. Designating channels

As discussed above, a channel is the partition (the target and projectile) and the set of quantum numbers needed to completely designate the incoming state. Our scheme is shown in figure 25. Therefore, at the very least a `<channel>` element must contain the quantum numbers of the state and the target and projectile. Because we anticipate reconstructing the resonances back into pointwise cross sections for plotting among other things, we should also connect to the `<reaction>` element (or a least its designator). For backwards compatibility with ENDF, we also need the MT and Q value.

Now, because resonances are a specific quantum state in a compound nucleus, not all channels can couple to a particular state. Therefore it makes sense to group together channels with similar quantum numbers into "spin groups" as in ENDF. These leads to denser tables of resonances and so more efficient storage. As in ENDF, we recommend adding a `<spinGroup>` markup containing all `<channel>`s with a common J and Π.

Requirement 34: `<spinGroup>`

34.1 The data structure shall have a spot for the J and Π quantum numbers
34.2 The data structure shall have a list of `<channel>`s
34.3 A flag to denote that a spin group is for potential scattering only.
34.4 For non-potential scattering spin groups, eliminated gamma channels with non-zero resonance widths are needed to ensure that the R-matrix does not have poles on the real axis.

Both the RRR and the URR should share the same master channel list. This aids in reconstruction since the number and kind of channels do not change with energy unless a threshold opens up. Also, to aid in reconstruction and complete specification of the "box" that defines the R-matrix, we must specify the boundary parameter B_c and the box radius a_c. We comment that in some older evaluations, the scattering length a_c is energy dependent and this is used as another mechanism to adjust the potential scattering contribution to the resonances.

Conventional R-matrix approaches for neutron channels assume that the outgoing waves are free waves hitting a hard-sphere. Newer approaches to the R matrix use alternative outgoing waves, for example Gurbich's (Gurbich, 2014)) uses optical model distorted waves. In addition, some channels (outgoing gamma and fission in particular) may use phenomenological shifts, penetrabilities and phases. Therefore we should allow the evaluator to override the phase, shift and penetrability factors.

Requirement 35: `<channel>`

35.1 The reaction designator; for resonances, this also specifies the "partition" (see Lane and Thomas (Lane, 1958)) III.C.4. It is expected that this designator maps correctly onto one in the `<reactions>` list, otherwise there may be problems when reconstructing resonances. The reaction designator can also specify the excited state of the residual nucleus or the other ejected particle.
35.2 The identities of the two particles in this channel. This can be done here or with an associated `<particlePair>` markup such as done in ENDF.
35.3 The ENDF MT, if applicable (deprecated)
35.4 All `<channel>`'s must have a link to a `<reaction>` in the outer `<evaluation>`'s `<reactions>` tree so that it is explicit where the data for this channel will be placed after the resonances are reconstructed. This `<reaction>` could be some sort of empty placeholder.
35.5 All quantum numbers needed to uniquely specify the reaction, this is needed for resonances as well. In particular, the spin s of the channel, the orbital angular momentum l, the total angular momentum J and any other quantum numbers. This eliminates a degeneracy implicit in the ENDF format and helps with the quality assurance of the data.
35.6 A flag to denote that a channel is "eliminated" and one should use the Reich-Moore approximation when dealing with resonances in this channel. This is very nearly the default for the radiative capture channels.
35.7 A flag to denote that a channel is a fission channel. This is not a two-body reaction, but can be treated as one for computing cross sec-

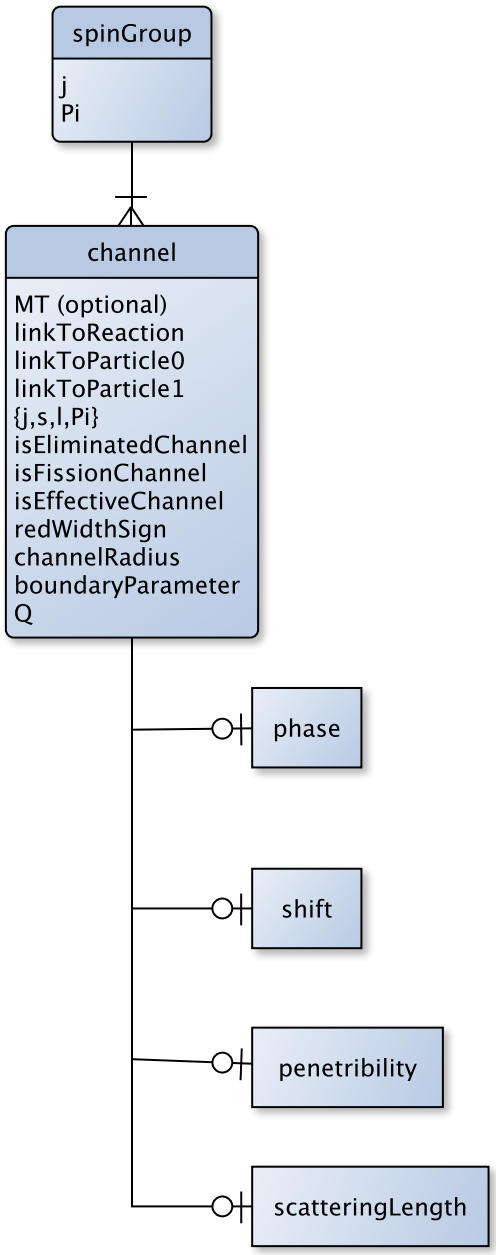


FIG. 25 Our proposed `<channel>` element. The two particles listed here could optionally be stored in a separate `<particlePair>` markup, provided that the particle pair is correctly associated with the channel.

tions.

35.8 A flag to denote that a channel is an "effective" one, such as an ENDF "competitive channel". This channel steals strength from the other channels, but should not be used for computing anything.

35.9 List the spin s for each resonance (resolves an ENDF ambiguity).

35.10 Optional boundary parameter B_c which, when specified, overrides the defaults in the ENDF-6 format

35.11 Sign of reduced width

35.12 Optionally the Q value of the reaction as it can help the evaluator. As this can be derived from the reaction products, it is strictly not needed as it is redundant.

35.13 To override the defaults, optionally specify

35.13.1 Phase $\varphi_c(E)$

35.13.2 Shift $S_c(E)$

35.13.3 Penetrability $P_c(E)$

35.13.4 Hard-sphere (channel) radius a_c (with potential dependence on energy). Likely need to be able to break it into multiple regions so that, for example, the RRR can have a constant one while the URR can have an energy dependent one. In ENDF, this is the "true radius".

35.13.5 The "effective radius", used in legacy evaluations for calculation of the phase.

SUGGESTED TEST: Check that spin groups and channels are consistent between the RRR, URR and channel specifications.

Discussion point:

We will need tests to ensure consistency between the `<channel>`s, the `<reaction>`s in the `<reactions>` and between the `<channel>`s and the `<RRR>` and `<URR>` columns.

Discussion point:

Is our approach enough to handle $(*, \gamma f)$ and/or fission reactions through class II states?

Discussion point:

Would user-definable (possibly fake) quantum numbers be useful? We would then need to define whether to combine using angular momentum adding rules (for Blatt-Biedenharn) or incoherently. For deformed nuclei, the K quantum number may require this feature.

Discussion point:

The channel wish list is very big. That said, a `<channel>` has all of the attributes of the `<reaction>` element. Does it make sense to completely separate the `<channel>` concept of the `<reaction>` element?

Resolution:
No, our <reaction> concept is very broad compared to the specific nature of the <channel> concept. Nevertheless, some codes that use these data may find it useful to design a parent class from which the channel and reaction classes might inherit.

Discussion point:
For charged particle channels, it is important to specify the correct mass and ionization state of the target atom so that electron screening and target recoil can be properly accounted for. How do we handle targets in a plasma environment?

Resolution:
Modeling targets in a plasma environment depends on additional parameters (density, temperature, etc.) beyond the scope of a nuclear database. This database may be able to provide mass and ionization state, but additional treatment must be left up to the end user.

2. Resolved resonances

The ENDF format supports several different approximations to the collision matrix from R matrix theory in Eq. (3). In addition, we should support the addition approximations discussed by Fröhner (Froehner, 2000) and in CALENDF (Ribon, 1986):

1. Pure potential scattering with either hard sphere or tabulated energy and/or ℓ-dependent scattering radius. Allows cross section and angular distribution calculation.
2. Single Level Breit Wigner (SLBW) approximation with one resonance. Allows cross section and angular distribution calculation. Only fission, capture, and elastic resonances are stored.
3. ENDF style SLBW. Allows only cross section calculation. Only fission, capture, and elastic resonances are stored. (deprecated)
4. Multi Level Breit Wigner (MLBW). Allows cross section and angular distribution calculation. CALENDF refers to this approximation as the Multi-Niveau Breit-Wigner (MNBW) format (Ribon, 1986). Only fission, capture, and elastic resonances are stored.
5. ENDF style MLBW. Allows cross section and angular distribution calculation for elastic reactions. Other channels are calculated using ENDF style SLBW cross sections. Only fission, capture, and elastic resonances are stored.
6. Reich-Moore. Allows cross section and angular distribution calculation. Gamma channels are eliminated and only fission, capture, and elastic resonances are stored.
7. ENDF style Reich-Moore. Allows cross section and angular distribution calculation for non-capture and fission channels. Capture and fission channel resonances may be used for cross section calculations only.
8. Full R matrix. Allows cross section and angular distribution calculation for all channels.

These approximations all require resonance energies E_λ and resonance widths $\Gamma_{\lambda c} = 2P_c\gamma^2_{\lambda c}$.

Discussion point:
Support storing resonance parameters from Brune transform, as described in reference (Brune, 2002)?

Requirement 36: Resolved resonance region (RRR)

- **36**.1 The actual <table> of resonance parameters, organized by <spinGroup>. The simplest arrangement is in columns; We may need to also tabulate the link of the <table> column to the <channel> element.
- **36**.2 The table may contain the uncertainties on the parameters as separate columns. If one parameter's uncertainty is given in the table then all should be in the table.
- **36**.3 LMax (an NLS-like thing) to specify the maximum ℓ value to sum to so as to get the potential scattering correct
- **36**.4 Upper and lower energy limits delineating the range of applicability of the resolved resonance region.
- **36**.5 A flag to denote the approximation used in the interpretation of the resonance parameters. In ENDF's LRF=7 format, this is analogous to the KLRF flag. Supported approximations should include SLBW, both ENDF and CALENDF style MLBW, ENDF and Fröhner style Reich-Moore, and full R-matrix.
- **36**.6 All background R matrix options (denoted by KBK in the ENDF format).

Discussion point:

In the ENDF LRF=7 format, one is to list each spin group that an evaluator uses in their evaluation. The problem with this approach is that it is difficult to assess whether the potential scattering contribution to an individual cross section has converged. It is entirely possible for an evaluator to fit a cross section with an unconverged set of spin groups. Were an evaluator to do this, the resonance parameters in the fit would essentially be tuned to reproduce both the resonant behavior and the potential scattering behavior of a cross section. The resulting parameters would then be unusable for the calculation of outgoing angular distributions.

Discussion point:

Although the Adler-Adler approximation is not currently used in major data libraries and its use in the ENDF format is deprecated, the approximation offers some practical advantages for efficient Doppler broadening. Similarly, the multipole expansion of resonances could offer similar gains. If the use of either increases, we will revisit the inclusion of features that support these.

Discussion point:

There are two complementary approaches to expressing the R-matrix: Kapur-Peiers and/or Wigner-Eisenbud. Both approaches use different boundary parameters B_c. They are mathematically equivalent, but the RRR approximations in ENDF all use Wigner-Eisenbud formulation. Should we support Kapur-Peiers as well?

Resolution:

No, because in Kapur-Peiers, one sets the boundary constant $B_c = L_c$. This leads to a complex pole E_λ, forcing us to mix data types (`complex` vs. `float`) in the `<table>` element in the `<RRR>` element. Other than this and charged particle scattering, there is no reason to include a `complex` data type.

Discussion point:

Fröhner (Froehner, 2000) suggests storing width amplitudes $\gamma_{\lambda c}$ instead of widths. This avoids sign confusions, they are not energy dependent and they do not vanish at threshold.

Resolution:

To do so would require us to denote the penetrabilities for all channels. For neutron, gamma and charged particle channels, the penetrabilities are known. For fission and competitive channels, the notion of a penetrability is nonsense. One could tabulate effective penetrabilities in the `<channel>` such that $\Gamma_{\lambda c}$ comes out right.

Discussion point:

The Brune transform (Brune, 2002) can be used as an alternate formulation of the R-matrix. The equations are somewhat different than those presented here or in the ENDF manual, but functionally they are equivalent and they remove the need to specify the boundary constant B_c.

3. Unresolved resonances

Above the RRR, we know that there are still distinct resonances, but they are too close together to be experimentally resolved. In this Unresolved Resonance Region (URR) we only know statistical properties of the resonances. Therefore, what is stored is not the resonance parameters, but ensemble averages of them. In the ENDF format (Trkov, 2009), the resonances are assumed to be in the SLBW approximation before averaging leading to the particular parametric form of the cross sections in the ENDF manual. However, CALENDF and other codes can use other parameterizations. Therefore a flag denoting the approximation of the R-matrix to use should be given.

Even though all we know are parameter averages in the URR, there is enough information to not only compute the average cross sections, but also the cross section probability distributions $P(\sigma_x|E)$ for all $x \in [\gamma, \mathrm{el}, \mathrm{tot}, f, ...]$. Both the PURR module in NJOY (MacFarlane, 2012) and the PURM module in AMPX (Dunn, 2002) can compute these cross section probability distributions which we term PURR tables. Upon reconstruction, the PDF for the PURR tables should get placed in the appropriate `<reaction>` as derived data.

Requirement 37: Unresolved resonance region (URR)

- **37**.1 Upper and lower energy limits delineating the range of applicability of the unresolved resonance region.
- **37**.2 Need number of degrees of freedom associated with each channel in the channel listing

37.3 Need a `<table>` of URR parameters, organized by `<spinGroup>`. This table must include columns for incident energy, mean level spacing, average widths for all channels.
37.4 ENDF assumes SLBW, allowing the construction of average cross section and PURR tables. This is a somewhat arbitrary restriction that is removed in CALENDF (Ribon, 1986). This URR data structure should allow all approximations that are supported for the RRR.
37.5 An `<axis>` and interpolation details to determine how to interpolate in incident energy among the average parameters.
37.6 All background R matrix options (denoted by `KBK` in the ENDF format).

Discussion point:
As one goes up in energy, one starts missing resonances little by little until one gives up and declares the URR region. The transition from fully knowing the RRR to fully NOT knowing the RRR (hence the URR), is not as abrupt as we would like. Should we add a table of estimated number of missing resonances as a function of energy and channel?

Discussion point:
Can we store Hauser-Feshbach transmission coefficients T_c's as the widths (if we additionally store the penetrabilities, shifts and phases)? If so, then we could use the Feshbach-Kerman-Koonin approach in-line for pre-equilibrium particle emission. This might be an interesting option in the overlapping resonance and fast regions.

4. Correcting reconstructed cross sections and distributions

Inevitably, either because of deficiencies in the approximation used, because of resonances missed in the evaluation process, or simply because of the tails of distant resonances, we must take action to correct the reconstructed cross section and angular distributions.

It is possible to make some corrections directly to the R-matrix. The easiest approach is just to add fake resonances to the resonance region just outside the energy range of the RRR. The tails of the fake resonances serve to correct the shape of the R-matrix inside the energy range of the RRR. This trick is commonly used to recreate the $1/v$-like behavior of the neutron capture cross section below thermal energies. Alternatively, one can add a background R matrix to mimic the effects of distant resonances (replacing $R_{cc'} \rightarrow R_{cc'} + R_c^{back}\delta_{cc'}$). One can do this with a complex-valued energy dependent function for $R_c^{back}(E)$ or with parameterizations provided by Fröhner (Froehner, 2000) or given in the SAMMY manual (Larson, 2006).

Requirement 38: Background R matrix

38.1 The `<backgroundRMatrix>` element shall have an option flag to denote the background R-matrix correction scheme, equivalent to ENDF's `KBK` flag.
38.2 For `KBK=0`, only need dummy resonances so no other information needed
38.3 For `KBK=1`, the `<backgroundRMatrix>` contains a set of complex-valued `<interp2d>` table for $R_c^{back}(E)$, one for each channel
38.4 For `KBK=2`, the `<backgroundRMatrix>` contains a set of eight parameters for each channel. These parameters are needed for SAMMY's logarithmic parameterization (see (Larson, 2006; Trkov, 2009)).
38.5 For `KBK=3`, the `<backgroundRMatrix>` contains a set of four parameters for each channel. These parameters are needed for Fröhner's parameterization (see (Froehner, 2000; Trkov, 2009)).

Discussion point:
Should we flag fake resonances beyond what is done to correct the R-matrix? Namely, should there be additional flags for resolved resonances in the RRR that appear as real resonances.

Many of the resonance approximations have severe shortcomings, in either the shape of the reconstructed cross sections or in the fact that outgoing angular distributions of emitted particles are untrustworthy or nonexistent. Indeed, one can never reconstruct the angular distributions of any emitted particles from fission resonances. Therefore, we need multiple schemes to amend or fix the reconstructed data tables from the RRR and URR. In figure 26 we provide an overview of more elaborate ways to fix reconstructed data.

In figure 26 we show a `<backgroundReaction>` element that contains all the things needed to fix the reaction linked to in the `linkToReaction` attribute. Additionally, the `addOrReplaceFlag` instructs the person processing the resonance data whether to add the cross section within `<backgroundReaction>` to the resonances or to replace the resonances entirely (this is useful for some older data). To accommodate for missing or mishandled outgoing particle data, we also add a `<reactionProducts>` element. Since several of the ENDF resolved resonance approximations DO NOT support angular distributions, the ENDF format provides a

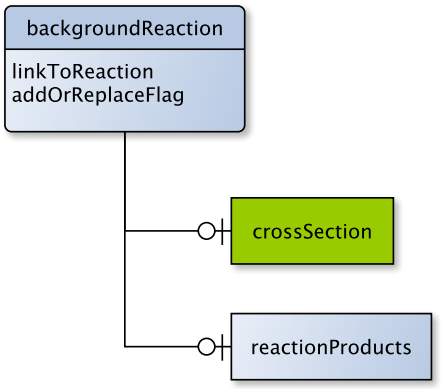


FIG. 26 Overview of elements to repair or amend cross section and other reaction data.

mechanism to store the "background" angular distributions separately. These "background" angular distributions naturally go in the `<reactionProducts>` element of the `<backgroundReaction>` element. Upon the reconstruction of the resonance region into pointwise data, the reconstructed data should point back to both the resonance data and these background data.

Requirement 39: The `<backgroundReaction>` element

39.1 Background cross-sections should not be given for (n,tot) and should be associated with the actual reaction to which the background cross-section is added. Otherwise we are adding a potential source for inconsistencies (background partial cross-sections not summing to the background total cross-section) or double counting.

39.2 The background cross sections must be associated with the resonance region for which they are the background. The obvious place is in the `<crossSection>` element of the `<backgroundCrossSection>` element.

39.3 The `<backgroundReaction>` shall have a flag to denote whether the cross section data within the element is meant to be added to the reconstructed resonances or replace them.

39.4 The `<backgroundReaction>` shall have a link to the reaction that these data will go to upon reconstruction.

39.5 The `<backgroundReaction>` shall have an optional `<reactionProduct>` element to store all of the outgoing particle data not reconstructed from the resonances associated with this reaction in the resonance energy range.

The background cross sections and related data should be kept with the RR, not the high energy file. The association between them is then explicit, eliminating a potential source of error. However, this is a different arrangement than in legacy ENDF where background cross sections are kept with the fast region cross section.

SUGGESTED TEST: Check for consistent energy ranges between RRR, URR and background reaction data

5. Expected resonance processing workflow

Given the changes to the organization of resonance and other nuclear data in the new structure, it is not unreasonable to ask what is the new resonance reconstruction workflow, where does reconstructed data go in the hierarchy once the data is reconstructed and how are the background data to be used?

To reconstruct the resolved resonance region, first reconstruct the cross sections from resolved resonance parameters, including the background R-matrix corrections, if given. Then for each reaction, add in the background cross sections for each reaction reconstructed. If it is possible to reconstruct angular distribution for this reaction (usually this is for neutron scattering only) and the distribution is not otherwise given, then reconstruct it, including background R-matrix corrections. Otherwise, take outgoing energy-angle data from appropriate reaction `<product>` or assume the angular distribution is isotropic so the full energy-angle distribution follows two-body kinematics.

Reconstructing the unresolved resonance region is somewhat different. If only the average cross sections are requested, reconstruct the average cross section from unresolved resonance parameters. If unresolved probability tables for cross section are requested, compute them using a processing code such as NJOY or CALENDF. For each reaction, add in background cross section for each (average) cross section reconstructed and add in the angle-energy distributions from appropriate reaction `<product>`. If the angle-energy distributions are not given, assume the angular distribution is isotropic and assume the energy dependence follows from two-body kinematics.

The reconstructed resolved and unresolved resonance regions are adjacent in incident energy and the unresolved region abuts the fast region. So, connect all the cross sections and distributions together for each reaction that was reconstructed. The reconstructed data go into the `<reaction>` element denoted in the `<channel>` element. The cross sections should be added as a new representation in the `<crossSection>` with appropriate links to denote what they are derived from. Distributions should be added as new representation to the appropriate `<product>` with appropriate links to denote what they are derived from. Unresolved probability tables, if computed, should be added to the `<crossSection>`s

as new derived data elements within each reconstructed `<reaction>`.

6. How to use the RRR elements in special circumstances

For resonance regions consisting solely of potential scattering, no special markup is needed. One simply must reconstruct the cross section using Eqs. (3-8) (or Eqs. (9-12) for angular distributions) and setting the R-matrix to zero. The outgoing wavefunction carries the potential scattering information. If one assumes hard-sphere, the scheme traditionally used in the ENDF format, the outgoing wavefunctions are given in the ENDF manual with the corresponding shifts, penetrabilities and phases. If one intends to use their own outgoing wavefunctions, the shift, penetrabilities and phases may be overridden in the `<channel>` element. Note, one must make sure to set the number of included channels (basically `Lmax` above or ENDF's NLS) to get the cross sections correct.

For evaluations that *only* give a scattering radius, there is no special markup. This is just the simplest form of potential scattering.

IV. COVARIANCE DATA

Users of nuclear data need covariance data to quantify uncertainty on the metrics of importance in their specific application. These metrics (such as k_{eff} in a criticality calculation) may have a deep dependence on the underlying nuclear data. Users mainly use deterministic group-wise methods (using the "Sandwich formula" below) or Monte Carlo techniques. Additionally, users need to visualize both the mean values and the uncertainty/-covariance on their data. We must do what we can to simplify all modes of covariance use.

The legacy ENDF format stores nuclear data covariance with a very complex set of schemes: the ENDF manual (Trkov, 2009) takes over 80 pages to describe seven distinct types of data. Arguably, there should be one "simple" data structure, after all a covariance matrix is, at its heart, just a matrix.

That said, we must deal with covariances not just within an observable, but across observables and evaluations. These covariances can also be quite large, far exceeding the size of the evaluations to which they refer. Therefore we must allow the storing of covariance data in several places, either with a particular data set (for covariance within a dataset), within an evaluation (for cross-reaction or cross observable covariance), or in an external file (for cross material or cross library covariance). In Fig. 27 we illustrate these points.

In the center of Fig. 27, we have one large covariance matrix composed of several blocks, each consisting of either a self- or cross- covariance for a schematic evaluation. Along the diagonal in orange are the self-covariances. A self-covariance is the covariance matrix for data with itself. These are stored in their corresponding data element, in this case a `<reaction>`'s `<crossSection>`. The cross-covariances between the different data set elements are shown in yellow. These connect to two different sets and so are not stored with the cross section data. Rather, they are stored in their own data tree in a `<covariances>` container. These blocks are not covariances in and of themselves. The blocks in white are a reflection of the yellow ones since the full covariance matrix is symmetric.

Requirement 40: Where to put covariance data

- **40**.1 Self–covariance data should be stored in the `<listOfDataRepresentions>` of a data element
- **40**.2 Cross–covariance data shall be stored in a `<covariances>` container, either within an evaluation or in an external file with `<covariances>` as the root node.
- **40**.3 `<covariances>` containers shall:
  - **40**.3.1 have a spot for optional documentation
  - **40**.3.2 allow one or more covariances

It is not unusual for an evaluator to produce an evaluation with correlated data: for example, the ratio of the capture cross section to the fission cross section was fit during the evaluation process. The fission and capture cross sections have a cross–reaction covariance in addition to their individual self–covariances. To ensure that this connection is noticed in practice, we require that all data correlated through cross–covariances be linked as illustrated in Fig. 28.

Requirement 41: Cross–covariance linkage

- **41**.1 Cross–covariance data shall have links to the element of the data correlated.
- **41**.2 The correlated data element will have links back to the cross–covariance data.

Discussion point:
In the future the number of cross-observable covariances may increase making the number of cross-covariance links grow to unmanageable size. We may need to revisit these requirements at some point.

Discussion point:
Should we enable storing confidence interval information? Confidence intervals can be computed for any PDF and therefore are more general than a covariance matrix.

Resolution:
Agreed. However the specific implementation is a low level data container issue.

Discussion point:
We should consider approaches for storing "uncertainty" data on variables that can take only discrete values, e.g. J^{Π}. The most obvious approach is to assign probabilities on each allowed value. The approach could be expanded to define discrete conditional probabilities on a series of correlated values.

A. Covariance definitions

Before proceeding to state the requirements, it is useful to review the properties of covariances and related objects.

When we measure a quantity x_i, we assume that we do not actually get the "true" value given by Nature, but rather one sample from a probability density function (PDF). We note that the vector of observables

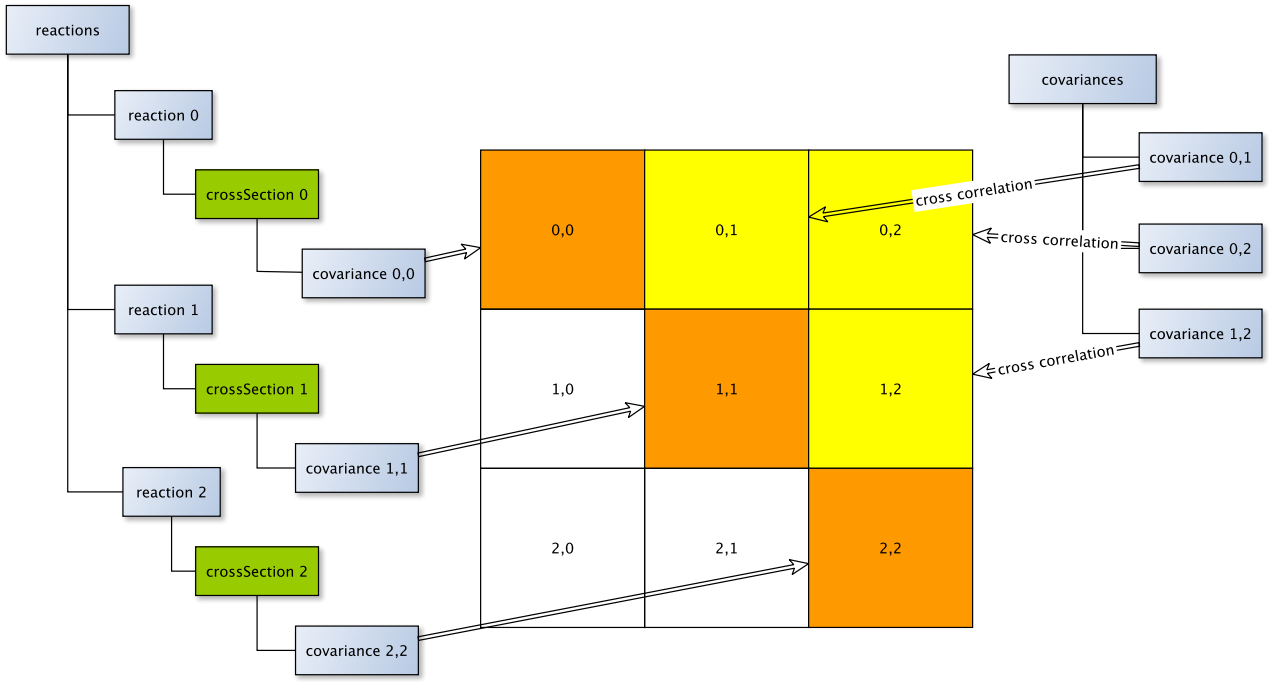


FIG. 27 Relationship of self-covariance and cross-reaction covariances, illustrating where they are to be stored in data files. The central covariance matrix is divided up into rectangular blocks, corresponding to different observables. The self–covariances are stored with the data elements in the tree on the left and the cross–covariance parts are stored in a separate list on the right.

$\vec{x} = (x_0, x_1, ...)$ do not need to be limited to a single observable within a reaction or even within an evaluation. Depending on the nature of the observable(s), the PDF might be Normal or Log-Normal (Zerovnik, 2013) or something else. For our purposes, we will assume that the PDF is either Normal or Log-Normal since the Central Limit Theorem guarantees that in the limit of large numbers of samples the peak of any PDF can be well approximated by a Normal distribution. We also include Log-Normal as an option since it forces values of an observable to be positive definite but otherwise behaves like a Normal distribution (Zerovnik, 2013).

For a quantity x_i, its PDF has an expectation value of $\langle x_i \rangle = \int dx_i \mathrm{PDF}(x_i) x_i$ and this would be stored in the ENDF file. The uncertainty on x_i is Δx_i. We define:

- **covariance**:

$$\begin{aligned} \mathrm{cov}x_{ij} &= (\Delta^2 x)_{ij} \\ &= \int dx_i dx_j \mathrm{PDF}(x_i, x_j)(x_i - \langle x_i \rangle)(x_j - \langle x_j \rangle) \\ &= \langle x_i x_j \rangle - \langle x_i \rangle \langle x_j \rangle \end{aligned} \tag{13}$$

- **relative covariance**:

$$\mathrm{rcov}x_{ij} = \mathrm{cov}x_{ij} / \langle x_i \rangle \langle x_j \rangle \tag{14}$$

- **uncertainty**:

$$\mathrm{unc}x_i = \sqrt{\mathrm{cov}x_{ii}} = \Delta x_i \tag{15}$$

- **relative uncertainty**:

$$\mathrm{runc}x_i = \Delta x_i / \langle x_i \rangle \tag{16}$$

- **correlation**:

$$\begin{aligned} \mathrm{corr}x_{ij} &= \mathrm{cov}x_{ij} / \mathrm{unc}x_i \mathrm{unc}x_j \\ &= \mathrm{cov}x_{ij} / \Delta x_i \Delta x_j \\ &= \mathrm{rcov}x_{ij} / \mathrm{runc}x_i \mathrm{runc}x_j \end{aligned} \tag{17}$$

Here, the covariance is a real, symmetric, positive definite $N \times N$ matrix. A covariance may be sparse or dense or even (band) diagonal.

> **Discussion point:**
> We comment that the covariance, uncertainty and correlations are all defined in a way independent of the underlying PDF. In ENDF, only Normal (Gaussian) PDFs are used. It is not unreasonable to support Log-Normal distributions as well. Going beyond these two common PDFs is certainly something to consider, but at this point are probably not needed. That said, we should consider exploring the support for confidence regions.

In the ENDF format, covariances and relative covariances are used exclusively in its covariance formats in MF=31–40. Unfortunately, most users want either only

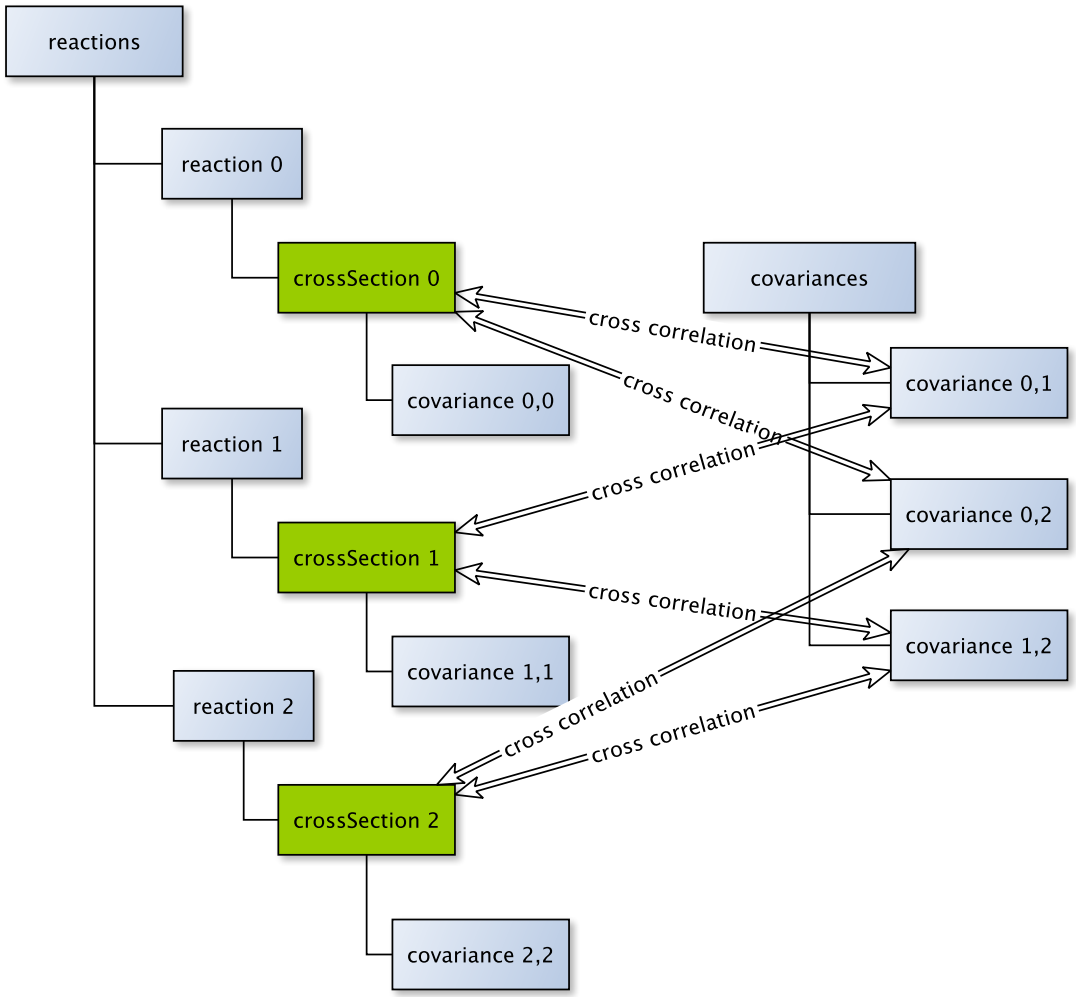


FIG. 28 Coupling between reaction data and the corresponding covariance. This figure is a continuation of the example in Fig. 27, highlighting the links between a data element and the cross-covariances associated with it.

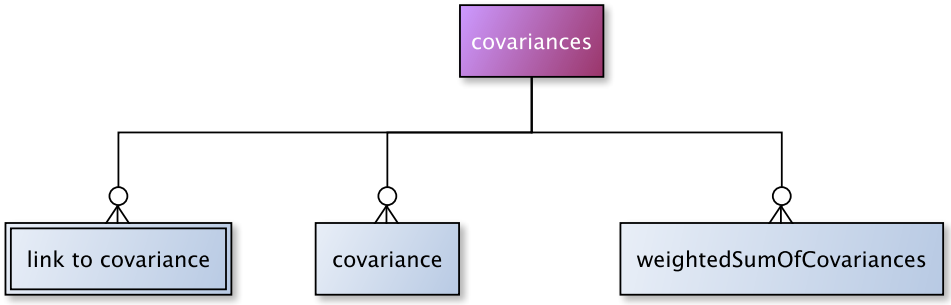


FIG. 29 The `<covariances>` element.

the (relative) uncertainty or the (relative) uncertainty and the correlation. Because of requirements **2**.2 and **2**.6, we really should attempt to accommodate both options within the same evaluation.

Returning to our example in Fig. 27, the self–covariances on the left side could be expressed several ways. They clearly could be expressed as either covariances or relative covariances. They could also be expressed as a combination of (relative) uncertainty and correlation matrices. In principal, both options must be

allowed. This is illustrated in section II.G in Fig. 10.

In Figure 10, the original data are "data version 0" and consist of an energy–cross section interpolation table. The main container, containing "data version 0" links to "covariance 0,0" where the covariances for these data are stored. From the data and the covariance, we can compute the uncertainty on the cross section data and generate a "data version 1" which now has an additional uncertainty column. "Data version 1" links back to "data version 0" and "covariance 0,0".

B. Covariance between different variables

There are many cases where an evaluator might want to store covariances on fitted parameters for a fitted function. Obvious examples include prompt neutron fission spectra given as e.g. a Watt spectrum or resolved resonance parameters. The covariance of this data may easily be stored in a matrix, provided we define the rule for mapping parameters to row/column in the covariance matrix. The `<grid>` element provides this mapping.

Requirement 42: `<grid>` elements

- **42**.1 Shall have an attribute denoting the number of elements
- **42**.2 For discrete variables, shall denote the packing order

C. Covariance of continuous functions

The covariance data we wish to store are sometimes discrete parameters. However, more often they are continuous functions of a variable such as incident neutron energy. To encode the covariance of a continuous function into a matrix, we must discretize. In the ENDF format, this is usually done by grouping in all the independent variables.

For example in ENDF, a cross section $\sigma(E)$'s covariance is given group wise as $\Delta^2\sigma_{ij}$. The group boundaries can be thought of as forming a basis function expansion where the basis functions are window functions:

$$B_i(E) = \begin{cases} 0 & E < E_i \\ 1/(E_{i+1} - E_i) & E_i \le E \le E_{i+1} \\ 0 & E > E_{i+1} \end{cases} \quad (18)$$

Thus, the basis function encodes the interpolation rule (in this case group-wise). To write the continuous covariance on the cross section $\Delta^2\sigma(E_1, E_2)$, we write

$$\Delta^2\sigma(E_1, E_2) = \sum_{ij} B_i(E_1)\Delta^2\sigma_{ij}B_j(E_2) \quad (19)$$

In ENDF, one other discretization method is used – the Legendre function expansion of angular distributions. In principal, many other discretization schemes are possible (various orthogonal function expansions, different order interpolation schemes, etc.)

To encode the discretization of a continuous function and to define the packing of covariance data as a function of independent variable, we require an element similar to the `<axis>` element discussed in section II.I. This container can also encode the packing rule for a list of discrete variables, such as resonance parameters. In keeping with the latest version of GND (Mattoon, 2012), we call this container a `<grid>` element:

Requirement 43: `<grid>` elements for continuous functions

- **43**.1 For grouped independent variables, shall contain the bin boundaries
- **43**.2 For other discretization schemes, shall denote the packing order of for example coefficients in an orthogonal function expansion

D. Covariance of multi-dimensional functions

In the previous section, we described how to discretize the covariance for a function of one independent variable. How might we handle a function with more than one independent variable? The easiest solution is to discretize in each independent variable separately. Each independent variable then has an associated `<grid>` element. We need only define the order we loop over elements in the individual grids to pack the covariance matrix.

We illustrate this with the case of the covariance on the Prompt Fission Neutron Spectrum (PFNS). The PFNS is a PDF $P(E'|E)$ with a covariance $\Delta^2\sigma(E'_1, E_1, E'_2, E_2)$. Let us discretize this:

$$\begin{aligned} &\Delta^2\sigma(E'_1, E_1, E'_2, E_2) \quad (20) \\ &= \textstyle\sum_{ij\ell m} B_i(E_1)B'_j(E'_1)\Delta^2\sigma_{ij\ell m}B_\ell(E_2)B'_m(E'_2) \end{aligned}$$

If we decide to say loop over i first then over j, we can effectively pack the i, j dependence into rows in the covariance matrix. Similarly, we can pack the ℓ, m dependence into columns.

So, we simply need a general rule for looping over the elements in a list of `<grid>`s. Here we state the simplest rule we can conceive. For a list:

```
<grid0>
<grid1>
<grid2>
...
<gridN>
```

Do the following:

- loop though highest index grid bins first (`<gridN>`)
- then the second last

- ...
- finally the first (<grid0>)

The pseudocode to read this appears as follows:

```
for i0 in grid0:
    for i1 in grid1:
        for i2 in grid2:
            ...
                for iN in gridN:
                    do something
```

Requirement 44: Multidimensional use <grid>

44.1 Shall have one <grid> per independent variable
44.2 Define a looping prescription

SUGGESTED TEST: Test that the number of rows/-columns is the product of the dimensionality of the <grid> elements making up the rows/columns

E. Covariance and correlation matrices

We are now in a position where we can define the structures that can be used to store self- and cross- covariance information. Both self- and cross- covariance information can be stored in <covariance> elements, either as absolute or relative covariances. The self-covariance can be converted into an uncertainty vector and a correlation matrix and the result will not introduce more couplings to other data. While a self-covariance is a full-fledged covariance in its own right, a cross-covariance is only a part of a larger covariance.

To store just the covariances, we define a <covariance> element whose structure is shown in Figure 30, panel (a). In this element, we note the presence of <row> and <column> elements. These elements connect the covariance to the underlying data being described and contain required <grid> elements described in the previous two sections. Because self–covariance data connects only one data set to itself, it has the same row <grid>s and column. On the other hand, cross–covariance data must have different rows and columns that depend on the data getting correlated. The requirements of a covariance are:

Requirement 45: <covariance> and <correlation>

45.1 The data structure shall have a spot for optional documentation
45.2 The data structure shall have a <row> element which includes a link to the original underlying data. The row must also be associated with the <grid> element to denote the data-to-cell in the matrix mapping. This association could be achieved with a link in the <grid> or by putting the <grid> within the <row> element.
45.3 For on-diagonal blocks of covariance, the <column> is equal to the <row>. For off-diagonal blocks, the <column> must also be specified in the same format as the <row> element.
45.4 The data structure shall have a <matrixData> element containing the matrix.
45.5 The element shall have a flag denoting whether the PDF for the data is Gaussian or Log-Normal
45.6 The element shall have a flag to denote whether the item described by the covariance is normalized or not.
45.7 A <covariance> element shall have a flag to denote whether it is a relative covariance.

Similarly, when one chooses to adopt the uncertainty and correlation matrix option, the requirements for a <correlation> are identical to that of a <covariance>. We give them separate tags to avoid confusion. We comment that there is no good reason why the uncertainty and correlation need to have <grid>s and <axis> elements that line up bin-by-bin. All that matters is that the domains of the independent variables are the same.

SUGGESTED TEST: Since covariances link to data and data links to covariance, we will need to check the hyperlinks for consistency.

In contrast to the arrangement given above in Fig. 30, GND 1.7 places the <grid> elements in what we would call the <matrixData> element. The GND 1.7 arrangement of elements is shown in Fig. 31. In GND 1.7, the <section> element corresponds to our <covariance> element and the covarianceMatrix element corresponds to our <matrixData> element. The benefit of the GND arrangement is that both the <gridded2d and <grid> elements are standard reusable low level data containers.

Within the covariance element is the matrix data itself and the matrix data element is shown in Figure 32. Its requirements are

Requirement 46: <matrixData>

46.1 Flag to denote whether this covariance is absolute or relative
46.2 Flag to denote that a covariance matrix represents a quantity that has a fixed normalization (e.g., a probability distribution such as $P(\mu, E)$ that must integrate to 1.0).
46.3 The data as a <matrix> or a <matrixSandwich> (see below)

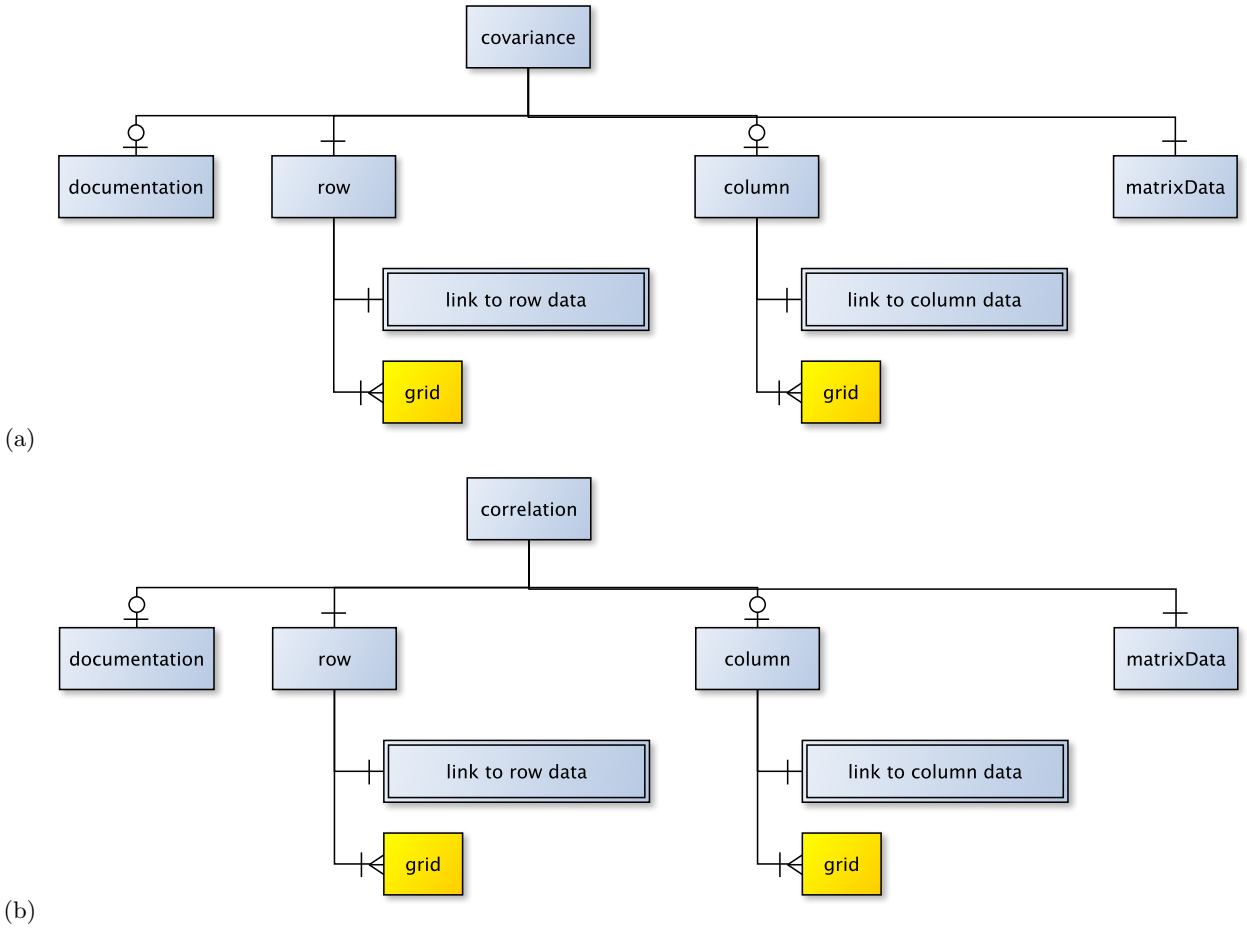


FIG. 30 An option for the `<covariance>` and `<correlation>` elements. As they share identical components, but have very different meanings, the element name must most likely be used to distinguish them.

F. Weighted sums of covariance

There are many times where a covariance can be written as the sum of two or more other covariances. The most common case in ENDF happens when the evaluator writes the covariance for the (n,tot) cross section as the sum of the (n,el) and (n, non-el) cross section covariances:

$$\Delta^2\sigma_{\text{tot}} = \Delta^2\sigma_{\text{el}} + \Delta^2\sigma_{\text{non-el}} \tag{21}$$

since $\sigma_{\text{tot}} = \sigma_{\text{el}} + \sigma_{\text{non-el}}$. In this case, the covariance for any one of the covariances can be written as a weighted sum of the others.

More generally, if a set of cross sections can be written as

$$\sigma = \sum_i w_i \sigma_i \tag{22}$$

then we may write

$$\text{cov}_{jk} = \sum_i w_i^2 \text{cov}_{jk}^i \tag{23}$$

These cases are easily addressed by adding a `<weightedSumOfCovariances>` element, shown in Fig. 33. This element takes advantage of the fact that the sum of two covariance matrices is still a covariance matrix. A `<weightedSumOfCovariances>` element should be usable anywhere where a `<covariance>` element can be placed. Incidentally, the ENDF format also allows for this construction. The requirements for this element are

Requirement 47: `<weightedSumOfCovariances>`

47.1 Numerical weights (in ENDF, these are just `floats` of a component
47.2 Either the components as `<covariance>`s or `links` to `<covariance>`s

By allowing us to sum up covariance data in this manner, we open ourselves up to new classes of data bugs. Therefore we suggest a few tests of these data:

SUGGESTED TEST: Testing for dead links is especially important here as one must *always* traverse all links in order to reconstruct a covariance.

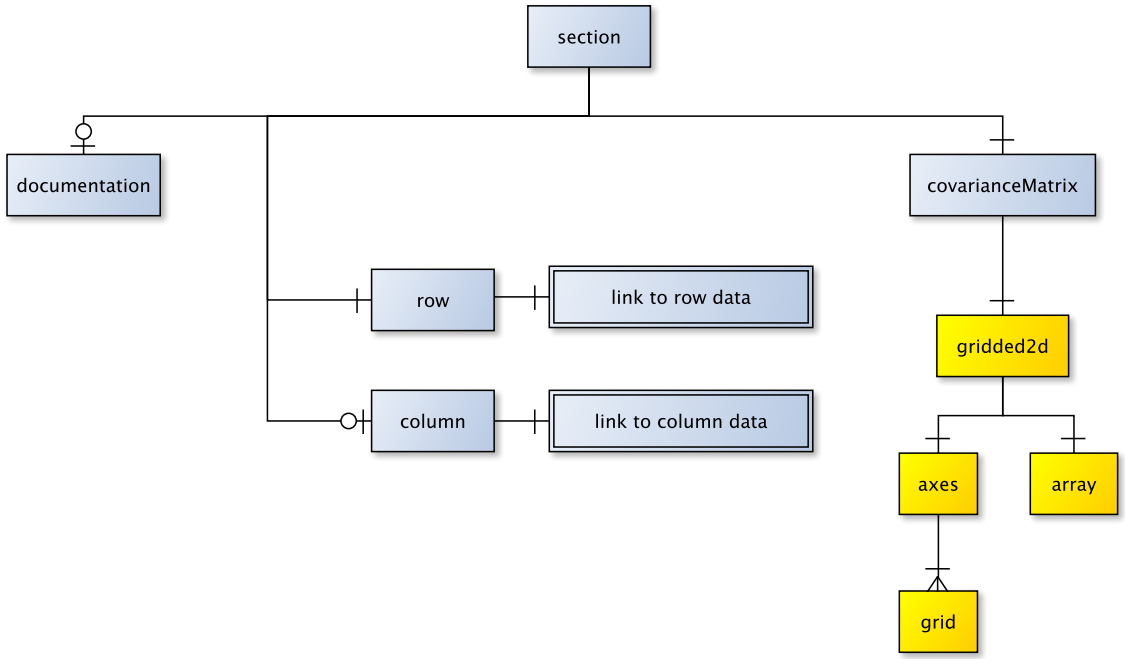


FIG. 31 The `<section>` element from GND 1.7 is analogous to our `<covariance>` element, but the `grid` element is contained in the low level `<gridded2d/axes>` element.

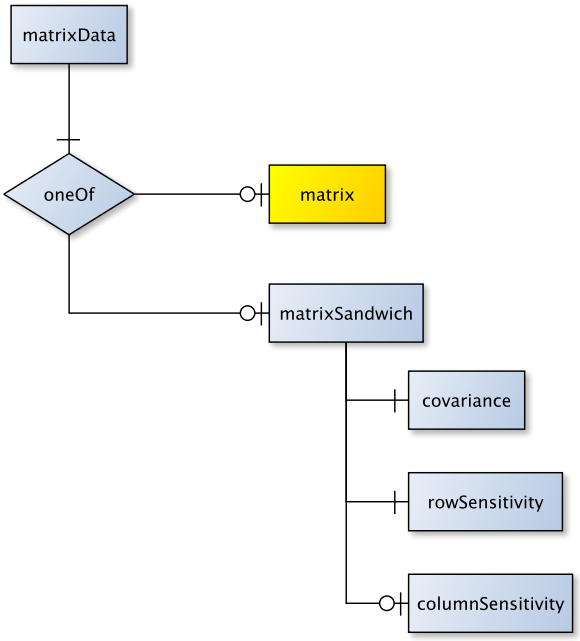


FIG. 32 Arrangement of parts of a `<matrixData>` element. The matrix may be stored either in low-level `<matrix>` container or as a "matrix sandwich" (see Eq. (25) below).

SUGGESTED TEST: All linked covariance data must be convertible to a `<covariance>` element so that they can be added.

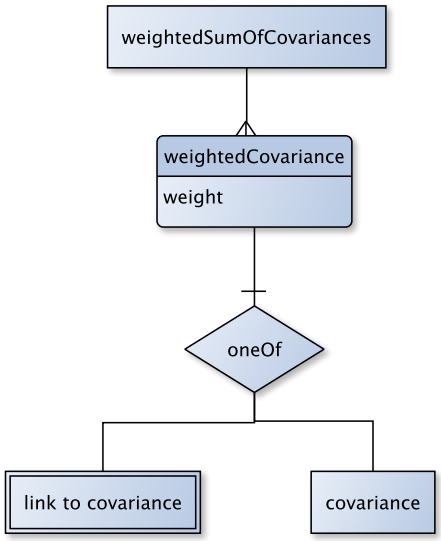


FIG. 33 The `<weightedSumOfCovariances>` element.

SUGGESTED TEST: A check for recursive linking leading to infinite loops and undefined covariance.

SUGGESTED TEST: All linked or explicitly stated covariances have identical limits on their `<row>` and `<column>` `<grid>` elements.

Discussion point:

Are we adding the potential for link bookkeeping errors by allowing the `<weightedSumOfCovariances>` construction for cross correlations? We have to require `link` matching between all parts of covariance.

G. The "Sandwich Formula"

Often times we have a parameter f_i that we want the covariance on, but it depends on something else, say $\vec{x}$, and it would be much more efficient to store the covariance on $\vec{x}$ directly. A case in point is the RRR parameters. The reconstructed cross section from tens of resonance parameters may have thousands of energy points to achieve a reasonable accuracy.

For function $\vec{f}(\vec{x})$, the $\text{sens}_{ij} == \partial f_i(\langle\vec{x}\rangle)/\partial x_j$ and is called the sensitivity matrix. Assuming that

$$f_i(\vec{x}) \approx f_i(\langle\vec{x}\rangle) + \sum_j \frac{\partial f_i(\langle\vec{x}\rangle)}{\partial x_j}(x_j - \langle x_j\rangle) \tag{24}$$

is a good approximation to the variation of $\vec{f}(\vec{x})$ around $\langle\vec{x}\rangle$, we can evaluate the covariance of f using the "sandwich formula" as

$$\text{cov}f_{ij} = \sum_{i'j'} \text{sens}_{ii'}\text{cov}x_{i'j'}\text{sens}_{j'j} \tag{25}$$

The "sandwich formula" can be reframed in terms of the relative covariance

$$\text{rcov}f_{ij} = \sum_{i'j'} \text{rsens}_{ii'}\text{rcov}x_{i'j'}\text{rsens}_{j'j} \tag{26}$$

where

$$\text{rsens}_{ij} = x_j \frac{\partial f_i(\langle\vec{x}\rangle)}{\partial x_j} = \frac{\partial f_i(\langle\vec{x}\rangle)}{\partial(\ln x_j)} \tag{27}$$

The "Sandwich Formula" provides the scheme for deterministic uncertainty propagation.

Discussion point:

In many cases, the sensitivity of model parameters can be precomputed. In this case, we may not need to store the sensitivity matrix itself. Should we allow this? It makes for smaller files, but shifts the burden of computing the sensitivities to the processing codes.

Resolution:

Yes, this is already the case for RRR parameters.

As an aside, the covariance admits an eigendecomposition into N eigenvalues λ_i with eigenvectors $\vec{v}_i$. The covariance can be diagonalized in the eigenbasis as

$$\text{cov}x_{ij} = \sum_k (\vec{v}_k)_i \lambda_k (\vec{v}_k)_j \tag{28}$$

This is the "Sandwich Formula" again, where the eigenvalues play the role of the sandwiched covariance matrix and the eigenvectors play the role of the sensitivity matrix. Often times the effective rank of a matrix N_{eff} is much smaller than the actual rank N because many of the eigenvalues are sufficiently close to zero that they may be neglected. The process of taking the main eigenvalues is called principal component analysis (PCA). Thus the "sandwich formula" storage scheme can be used to efficiently pack covariance matrices even in the absence of underlying parameter dependencies by using PCA. we will flesh this idea out further in the example below in section IV.I.

To support the "Sandwich Formula", we must define the structure of a `<matrixSandwich>` and a `<sensitivity>`:

Requirement 48: `<matrixSandwich>`

- **48**.1 The underlying parameter `<covariance>`
- **48**.2 A `<sensitivity>` for the rows of the covariance
- **48**.3 If the block of the matrix is off-diagonal, a `<sensitivity>` for the column as well.

Requirement 49: `<sensitivity>`

- **49**.1 Optionally a `<documentation>` element
- **49**.2 The `<matrixData>` for the sensitivity matrix
- **49**.3 A `<column>` with a `link` pointing to the `<column>` element's `<axis>` of the underlying parameter covariance matrix. This also defines the packing of the sensitivity matrix since we want them to match up for the matrix multiplication.
- **49**.4 A `<row>` that mimics the `row` one would get if we were storing the full covariance on the derived data. Therefore we need an `<axis>` element to determine the packing of the sensitivity matrix and a (possibly fake) link to the derived data.
- **49**.5 An option for precomputed sensitivity matrices in the resolved resonance region (to store the MT32 covariances in ENDF).

Discussion point:

It was requested by a member of the nuclear data community determine if it is possible to "break the covariance matrix out into sources", namely from different experiments or normalization. It is possi-

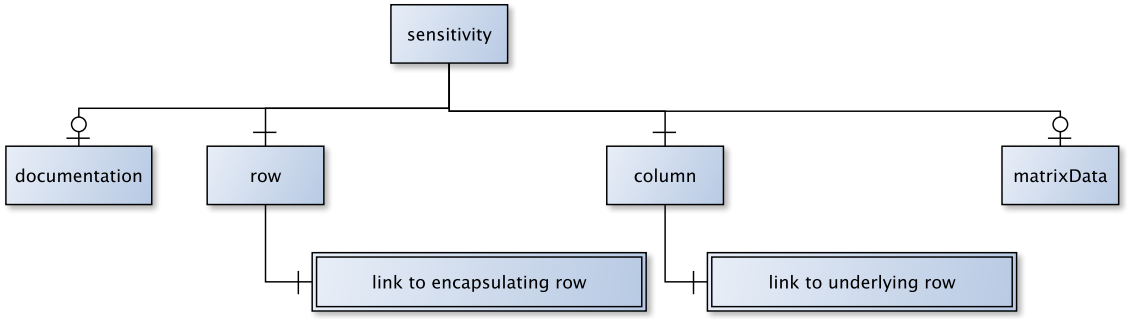


FIG. 34 The <sensitivityMatrix> element which connects the small matrix inside a "matrix sandwich" to the external covariance matrix in Eq. (25).

ble to do this (see for example Refs. (Brown, 2015) or (Neudecker, 2012) or many others). However, this amounts to storing *all* the data used in the evaluation process within the evaluation itself. Much of the underlying data covariance can be generated with the information in the main documentation block of an evaluation and the mean values should be contained in EXFOR. In the future we will have to explore what value is added by including this information with the covariance data in the evaluation.

H. Monte Carlo sampling

How can one use a covariance to generate realizations for a Monte Carlo approach to uncertainty quantification? Suppose we have some $\vec{x}$ with a Normal PDF $P(\vec{x})$ specified by the mean $\langle\vec{x}\rangle$ and covariance covx_{ij}. To find the expectation value of a function $f(x)$, we do

$$\langle f\rangle = \int d\vec{x} P(\vec{x}) f(\vec{x}) \approx \frac{1}{N}\sum_R f(\vec{x}_R) P(\vec{x}_R) \qquad (29)$$

Here the sum is over realizations of $\vec{x}$ drawn from the PDF. To generate the realizations R, we use principal component analysis (PCA) again:

$$\vec{x}_R = \langle\vec{x}\rangle + \sum_i \xi_R^i \vec{v}_i \sqrt{\lambda_i} \qquad (30)$$

Where ξ_R^i is drawn from a (log) normal distribution and λ_i and $\vec{v}_i$ are the eigenvalues and eigenvectors of the absolute covariance C. Another option is to do the principal component analysis on the relative covariance C_rel, and construct realizations as

$$\vec{x}_R = \langle\vec{x}\rangle * \sum_i \xi_R^i \vec{v}_i \sqrt{\lambda_i} \qquad (31)$$

Discussion point:

Should we support ensembles of evaluations or evaluation parts (like TMC or list-mode output)? Would need index of realizations maybe. Could this be handled using the `metaEvaluation` scheme?

Resolution:

One would need reasonable number of samples N_{samp} for each of the N_{dim} directions. This means that one needs $(N_{\rm samp})^{N_{dim}}$ samples to effectively sample all of $\vec{x}$'s PDF to reliably propagate uncertainty. Saving these samples is then not really an effective savings of space.

I. Examples of covariance data usage in this hierarchy

In order to illustrate some of the concepts we have developed to describe our implementation of covariance data, we turn to three examples.

In our first example, we demonstrate the storing of Prompt Fission Neutron Spectrum (PFNS) covariance. The situation is shown in figure 35. In this figure, the PFNS itself is stored in the fission <reaction> under the neutron reaction product. The PFNS has as a subelement the covariance data accompanying it. The <covariance> itself is a self–covariance, is relative, and each row / column of the matrix sums to 0 (due to the normalization condition on the spectrum). The element has documentation and <matrixData>. Since the covariance is on a PFNS, and the PFNS is a function of both the incoming and outgoing neutron energy, there are two <grid> elements associated with the <row> element.

In our second example, we show how to use the sensitivity matrix approach and the eigen-decomposition of a covariance to compress the covariance matrix. This is illustrated in figure 36. In this figure, we focus on a generic self-covariance covx_{ij}. The <documentation> and <row>

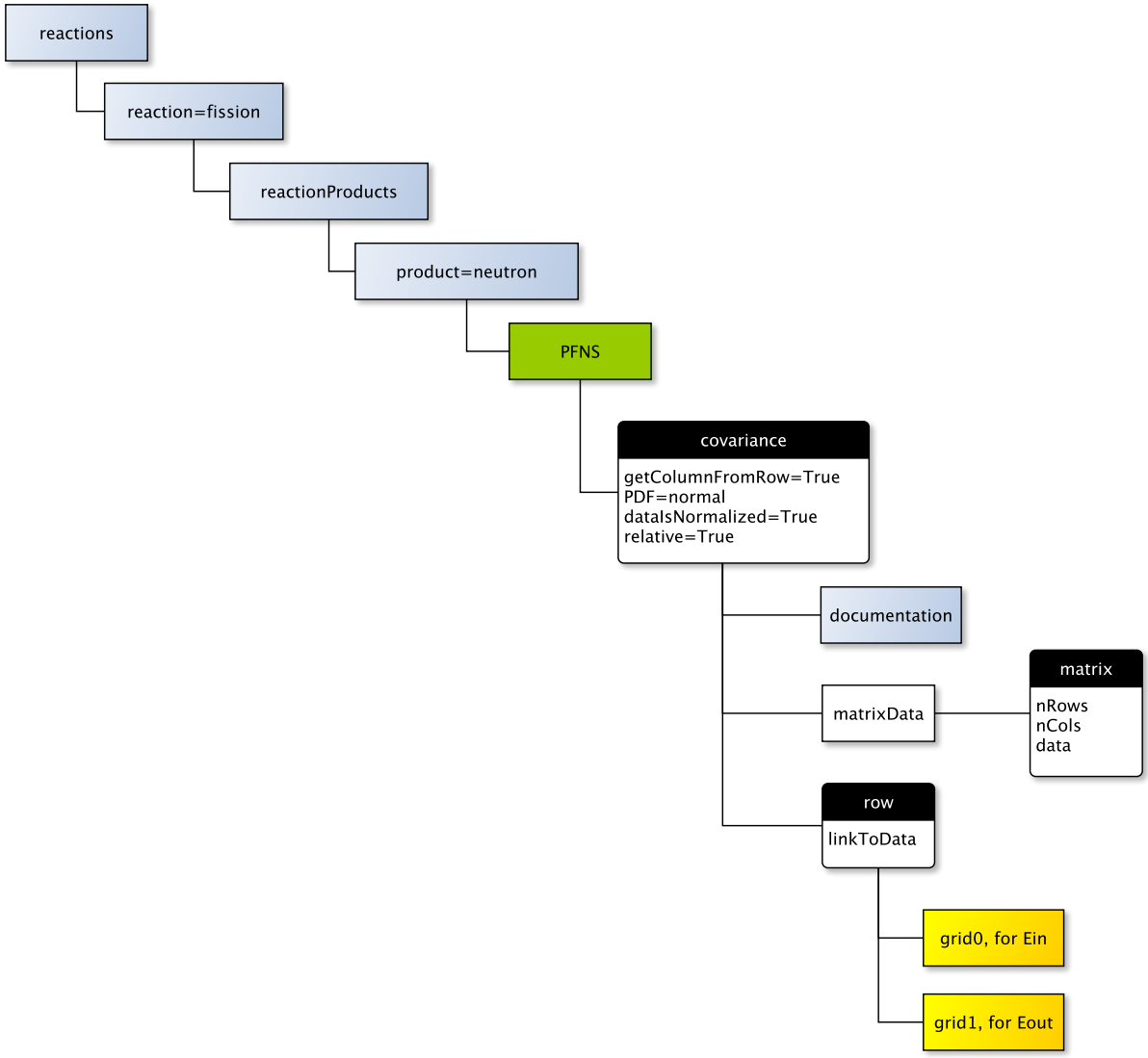


FIG. 35 A sample usage of covariance data demonstrating the layout of Prompt Fission Neutron Spectrum covariance.

data (including a `<grid>` specification and link to the original data) are shown on the left side of the figure. We focus on the covariance itself in a `<matrixData>` container. We diagonalize our generic self–covariance using equation (28), producing a list of eigenvalues $\{\lambda_k\}$ and associated eigenvectors $\{\vec{v}\}$. As we observed in the discussion around equation (28), equation (28) is already in the form of the "Sandwich formula". The sensitivity matrix is just the matrix of eigenvectors, $\text{sens}_{ik} = (\vec{v}_k)_i$ and the inner covariance is the matrix of eigenvalues $\text{cov}_{kk'} = \delta_{kk'}\lambda_k$. The matrix of eigenvalues goes in the inner `<covariance>` element in figure 36. The `<row>` (and `<grid>`) of the `<covariance>` is especially simple since the inner matrix is just a matrix of eigenvalues. The sensitivity matrix (the matrix of eigenvectors) goes in the `<sensitivityMatrix>` element and links to the inner and outer `<rows>`. We comment that the ENDF format also allows this approach to storing a matrix, using the MF=30 as described in subsection 30.3.3. To our knowledge, no one has used the MF=30 format in a production environment.

In our final example, we show how we re-implement the ENDF format's MF=32 parameterized resonance covariance data. This is illustrated in figure 37. In this example, we will use the `<sensitivityMatrix>` scheme again, but this time it will be a little simpler since the sensitivity matrix is known analytically and is given in the SAMMY manual (Larson, 2006). In figure 37, we again start from the upper left side with the outermost `<covariance>` element. In this case, the covariance matrix we refer to is the full covariance of all of the reconstructed pointwise cross sections. Because of that,

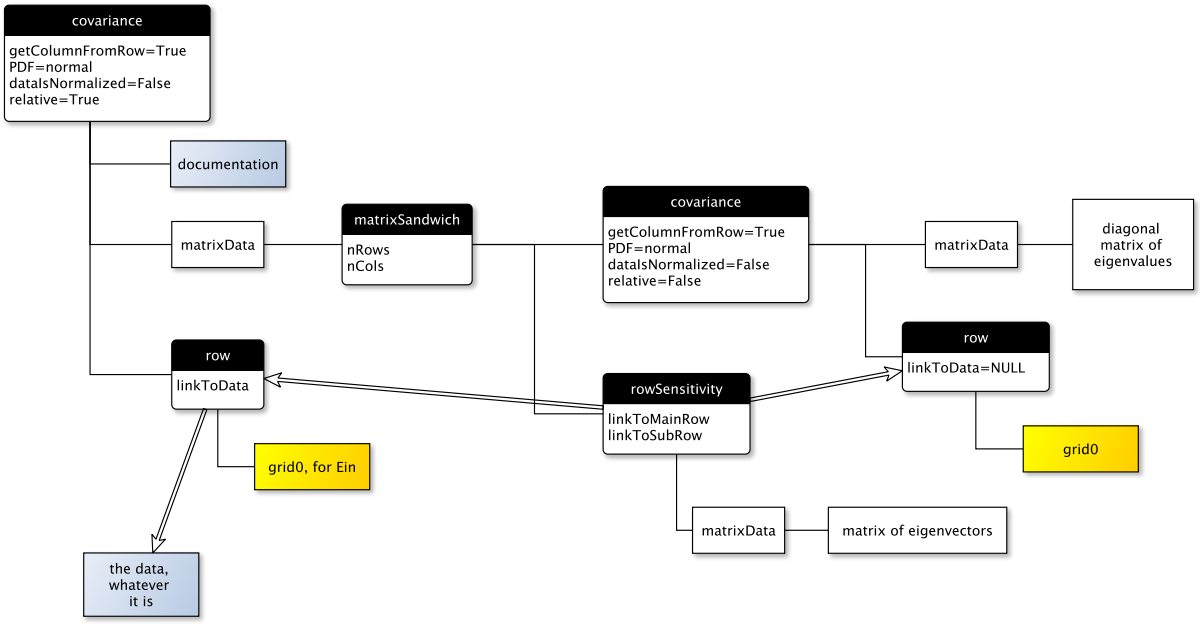


FIG. 36 Illustrating the use of the `<sensitivityMatrix>` approach to compress a covariance matrix using the principal components of the covariance determined from an eigenvalue decomposition.

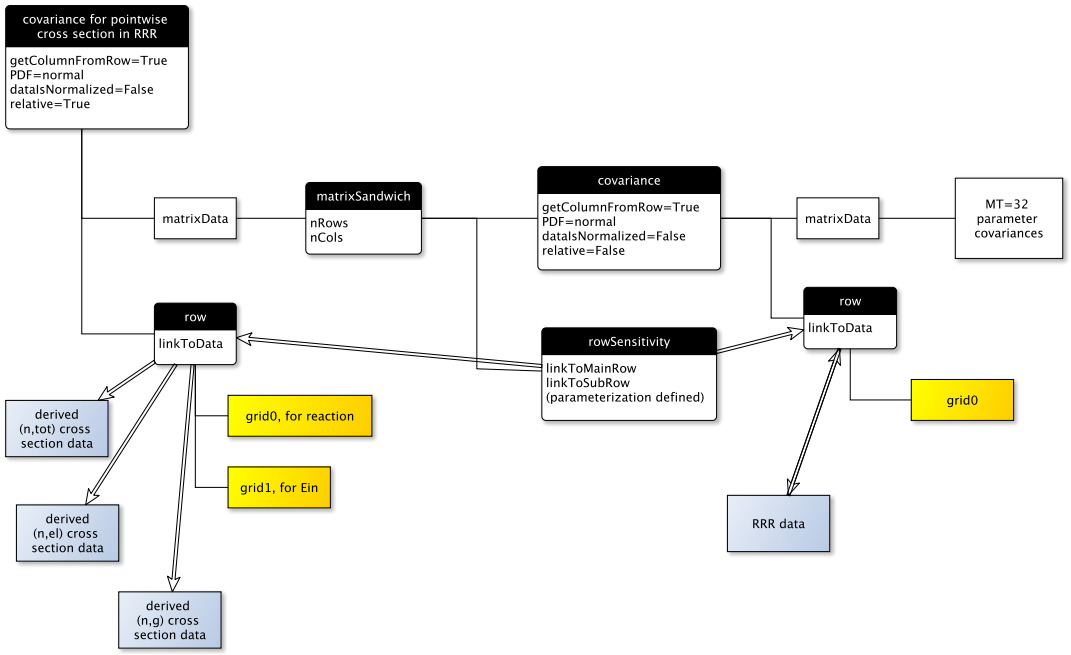


FIG. 37 Illustrating the storage of resonance covariance data. This is our approach to the ENDF format's MF=32.

the `<row>` element of the outer `<covariance>` points to several cross section data sets and has two `<grid>` elements, one for the incident energy grid and another to allow looping over the reactions themselves. Inside the `<matrixData>` and `<matrixSandwich>`, we again find the `<sensitivityMatrix>` and the innermost `<covariance>` containing the matrix of resonance parameter covariance. The `<row>` element in the innermost `<covariance>` element describes how resonance parameters are packed into the `<covariance>` element.

V. SPECIAL CASES

The ENDF/B-VII.0 and ENDF/B-VII.1 libraries (Chadwick, 2006, 2011) contains 14 separate sub-libraries covering a variety of reaction data types. All of these must be covered by the data structure whose requirements we are drafting. Several of the sub-libraries have special requirements that require further discussion:

- Atomic data (§V.A)
- Charged particle reactions (§V.B)
- Fission reactions (§V.C)
- Fission product yields (§V.D)
- Radiative capture (§V.F)
- Spallation (§V.E)
- Thermal scattering law data (§V.G)

A. Atomic scattering

The atomic scattering and relaxation sub-libraries in ENDF store data for electron and photon collisions with atoms as well as the atomic de-excitation data. Although the ENDF format is not meant to be a comprehensive resource for atomic reaction data, it has found a niche for storing data needed for electron and photon transport in otherwise neutron (or other nuclear) dominated transport applications.

More information on atomic data can be found on the Virtual Atomic and Molecular Data Center (VAMDC) at `http://portal.vamdc.org/`. The VAMDC is a consortium of atomic and molecular data centers covering a variety of data types including experimental, evaluated, bibliographic, structure and reaction data. The entire code system is detailed in `http://www.vamdc.org/standards`. The VAMDC data are stored in the VAMDC-XSAMS format (an XML format). The format is described at `http://www.vamdc.org/documents/standards/dataModel/vamdcxsams/index.html`. There is also a wiki describing the entire system (web infrastructure as well as format) at `http://voparis-twiki.obspm.fr/twiki/bin/view/VAMDC/PortalHelp`.

Atomic scattering data in ENDF includes only electromagnetic (electrons and gammas) projectiles interacting with the electronic orbitals of an atom. These data are very similar to nuclear reaction data, but simpler in some ways and more complex in others. The ENDF atomic scattering data are given in reactions specified by MT=500-599 in the formats specified by MF=23, 26, 27, 28. In the ENDF/B-VII.1 library (Chadwick, 2011), atomic data are collected in three sub-libraries:

- **atomic relax relaxation** (NSUB=6): Details the de-excitation of excited atoms or ions following an excitation or ionization event. The excitation process either excites an electron to a higher shell or knocks it out completely, resulting in a vacancy. This library tabulates the cascade of events that can occur as the atom or ion relaxes to its ground state. These data are handled by the particle properties database(s) discussed in section II.D and detailed in Ref. (WPEC Subgroup 38, 2015a).
- **electro-atomic scattering** (NSUB=113) Provides cross section and particle production data for electrons scattering off neutral atoms. Reactions tabulated include elastic scattering, bremsstrahlung and inelastic scattering resulting in either atomic excitation or ionization. We comment that electrons, being charged particles, do not have a total cross section nor an integrated elastic cross section. As they have no "hard" or nuclear elastic interactions, their elastic scattering cross section is analytic and given by the Mott cross section.
- **photo-atomic scattering** (NSUB=3) Provides cross section and particle production data for photons scattering off neutral atoms. Reactions include elastic scattering and inelastic scattering resulting in either atomic excitation or ionization.

Because the recoil of the target atom is small (and it would be difficult to compute the response of the electron cloud and nucleus in any event), it is neglected. Therefore, the data are considered to be given always in the lab frame and all scatterings transfer zero energy to the residual atom.

These data are given in a standard `<reaction>` element with the following additional requirements:

Requirement 50: Atomic reaction data

50.1 A standard `<reaction>` element whose outgoing particles are photons, electrons and/or a residual atom
50.2 A particle property database for atomic and ionic de-excitation (i.e., decay) data (WPEC Subgroup 38, 2015a)
50.3 Outgoing photons may optionally use form factors for coherent and incoherent photon scattering (see MF=27) in a `<dCrossSection_dOmega>` element. This is detailed below.
50.4 Usual outgoing distributions, with
  50.4.1 Electron and gamma multiplicity (yields)
  50.4.2 Outgoing electrons or photons may use form equivalent to LAW=1 (continuum, used for bremsstrahlung and ionization)

(same as MF=6, LAW=1), or

50.4.3 Outgoing electrons or photons may use form equivalent to LAW=2 (two-body elastic) (same as MF=6, LAW=2), or

50.4.4 Outgoing electrons or photons may use form equivalent to LAW=8 (energy transfer for excitation, used for excitation and bremsstrahlung), described in MF=26; if so use `<interp1d>` to tabulate the energy transfer $E_T(E)$ for LAW=8

50.4.5 The residual atom product element with a location for the fluorescence yield. This is typically a float with units eV/photoionization

50.5 For bremstrahlung, only the outgoing electron and its dE/dx need to be given. The outgoing photon spectrum and angular distribution (relative to the incident electron) can be computed.

50.6 An optional documentation element

50.7 Any links to covariance (if applicable)

The ENDF system for neutron and photon production data allows two alternatives for storing angular distribution data. One is by probability per unit $\cos(\theta)$ vs. $\cos(\theta)$, and the other is by Legendre coefficients. Neither of these is a "natural" method for photons. The natural method would be atomic form factors or incoherent scattering functions. These are discussed briefly below.

1. Incoherent photon scattering

The cross section for incoherent scattering is given by:

$$\frac{d^2\sigma_{\rm incoh}(E)}{d\mu dE'} = S(q,Z)\frac{d^2\sigma_{KN}(E)}{d\mu dE'} \ , \tag{32}$$

where:

$d^2\sigma_{KN}/d\mu dE'$**:** the Klein-Nishina cross section (Klein, 1929) which can be written in closed form.

$S(q,Z)$**:** the incoherent scattering function. At high momentum transfer (q), S approaches Z. In the other limit, $S(0,Z)=0$.

q**:** the momentum of the recoil electron (ENDF specifies this in Å^{-1}).

$$q = \alpha\left[1+\left(\frac{\alpha'}{\alpha}\right)^2 - 2\mu\left(\frac{\alpha'}{\alpha}\right)\right]^{1/2} \tag{33}$$

α**:** E'_γ/m_0c^2,

E'_γ**:** the scattered photon energy,

μ**:** $\cos(\theta)$.

The angular distribution can then easily be calculated. Values of $S(q,Z)$ are tabulated as a function of q. The user presumably will have subroutines available for calculating q for energies and angles of interest and for calculating Klein-Nishina cross sections. The user will then generate the cross sections for the appropriate cases by calculating q's, looking up the appropriate values of S, and substituting them in the above formula.

2. Coherent photon scattering

The coherent scattering cross section is given by:

$$\frac{d^2\sigma_{\rm coh}(E)}{d\mu dE'} = \pi r_0^2\left(1+\mu^2\right)\times \left\{\left[F(q,Z)+F'(E)\right]^2 + F''(E)^2\right\} \ , \tag{34}$$

where:

q**:** $\alpha\left[2(1-\mu)\right]^{1/2}$, the recoil momentum of the atom (ENDF specifies this in $\mathring{A}^{-1}$),

r_0**:** e^2/m_0c^2, the classical radius of the electron.

$F'(E)$**:** the real part of the anomalous scattering factor.

$F''(E)$**:** the imaginary part of the anomalous scattering factor.

The quantity $F(q,Z)$ is a form factor, which can be easily tabulated. At high momentum transfer (q), F approaches zero. In the other limit $F(0,Z)$ tends to Z. The anomalous scattering factors are assumed to be isotropic. In addition, they smoothly approach zero at 1.0 MeV and can be assumed to be zero at higher energies.

An alternative way of presenting the photon scattering data would be to tabulate incoherent scattering functions and form factors. Users could then provide processing codes to generate the cross sections from this information. The calculation is quite straightforward and allows the user to generate all this scattering data from a relatively small table of numbers. The incoherent and coherent scattering data should always be presented as scattering functions and form factors, respectively, whether or not data are included.

Requirement 51: Photo-atomic data

51.1 An `<interp1d>` element for the incoherent scattering function $S(q,Z)$.

51.2 An `<interp1d>` element for the coherent scattering function $F(q,Z)$.

51.3 A pair of `<interp1d>` elements for the real and imaginary parts, $F'(E)$ and $F''(E)$, of the anomalous form factor.

Discussion point:

An `<interp1d>` element supporting complex numbers could simplify these data.

B. Charged particle elastic scattering

As we outlined in subsection III.C, charged particles do not have a finite total cross section or angle integrated elastic cross section. Quantum mechanically, charged particle elastic scattering is a sum of Coulomb and nuclear amplitudes:

$$A = A_{\text{Coul}} + A_{\text{nucl}} \tag{35}$$

The Coulomb piece is analytic and well known. The nuclear piece must be evaluated. The cross section for elastic scattering is of course the square of the amplitude so the differential cross section has three terms:

$$\frac{d\sigma_{\text{el}}(E)}{d\mu} = \frac{d\sigma_{\text{Coul}}(E)}{d\mu} + \frac{d\sigma_{\text{int}}(E)}{d\mu} + \frac{d\sigma_{\text{nucl}}(E)}{d\mu} \tag{36}$$

The last two terms in Eq. (36) are traditionally lumped together in a "nuclear+interference" term. Note, this scheme is used for elastic scattering of electrons in the electro-atomic sub-library in ENDF, but with no nuclear amplitude.

Here the Coulomb cross section is different for distinguishable (Cd) versus identical (Ci) particles:

$$\frac{d\sigma_{Cd}(E)}{d\mu} = \frac{\eta^2}{k^2(1-\mu)^2} \tag{37}$$

$$\frac{d\sigma_{Ci}(E)}{d\mu} = \frac{2\eta^2}{k^2(1-\mu^2)}\left[\frac{1+\mu^2}{1-\mu^2} + \frac{(-1)^{2s}}{2s+1}\cos\left(\eta\ \ln\frac{1+\mu}{1-\mu}\right)\right] \tag{38}$$

Here s is the spin of the identical particles.

Whether the target and the projectile are identical or not, the Coulomb term is very singular:

$$\frac{d^2\sigma_{\text{Coul}}(E)}{d\mu} \propto \frac{\eta^2}{k^2(1-\mu)^2} \tag{39}$$

Therefore, the elastic cross section diverges at small incident E and small angles ($\mu \to 1$). One might think that, since this is analytic, we don't have to store it and there is no problem. The problem is that since the Coulomb amplitude carries the square-root of these divergencies, the interference term σ_{int} in the total elastic differential cross section also carries divergencies.

The ENDF format provides two solutions:

1. Use a Legendre series expansion of the "nuclear+interference" term or

2. use a "fake cross section" representation of the "nuclear+interference" term.

1. Legendre series expansion approach:

The Legendre series expansion can be found by writing the net elastic scattering cross section for distinguishable particles as:

$$\frac{d\sigma_{ed}(E)}{d\mu} = \frac{d\sigma_{cd}(E)}{d\mu} - \frac{2\eta}{1-\mu}\text{Re}\left\{\exp\left(i\eta\ln\frac{1-\mu}{2}\right)\sum_{l=0}^{\text{NL}}\frac{2l+1}{2}a_l(E)P_l(\mu)\right\} + \sum_{l=0}^{2\text{NL}}\frac{2l+1}{2}b_l(E)P_l(\mu) \tag{40}$$

and the cross section for identical particles as:

$$\begin{aligned}\frac{d\sigma_{ei}(E)}{d\mu} &= \frac{d\sigma_{ci}(E)}{d\mu} \\ &- \frac{2\eta}{1-\mu^2}\,\mathrm{Re}\left\{\sum_{l=0}^{\mathrm{NL}}\begin{bmatrix}(1+\mu)\exp\left(i\eta\ln\frac{1-\mu}{2}\right)\\ +(-1)^l(1-\mu)\exp\left(i\eta\ln\frac{1+\mu}{2}\right)\end{bmatrix}\frac{2l+1}{2}a_l(E)P_l(\mu)\right\} \\ &+ \sum_{l=0}^{\mathrm{NL}}\frac{4l+1}{2}\,b_l(E)P_{2l}(\mu) \end{aligned} \tag{41}$$

where the a_l are complex coefficients for expanding the trace of the nuclear scattering amplitude matrix and the b_l are real coefficients for expanding the nuclear scattering cross section. Both a_l and b_l should be related, but ENDF lets them be independently chosen.

It is also possible to suppress the divergence in the interference term with an alternative definition of the Legendre moments. Writing

$$\frac{d\sigma_{Rd}(E)}{d\mu} = (1-\mu)\left[\frac{d\sigma_{ed}(E)}{d\mu} - \frac{d\sigma_{cd}(E)}{d\mu}\right] \tag{42}$$

and

$$\frac{d\sigma_{Ri}(E)}{d\mu} = (1-\mu^2)\left[\frac{d\sigma_{ei}(E)}{d\mu} - \frac{d\sigma_{ci}(E)}{d\mu}\right] \tag{43}$$

Then σ_R can be given as a Legendre polynomial expansion in the forms:

$$\frac{d\sigma_{Rd}(E)}{d\mu} = \sum_{l=0}^{\mathrm{NL}}\frac{2l+1}{2}c_{ld}(E)P_l(\mu) \tag{44}$$

and

$$\frac{d\sigma_{Ri}(E)}{d\mu} = \sum_{l=0}^{\mathrm{NL}}\frac{4l+1}{2}c_{li}(E)P_{2l}(\mu) \tag{45}$$

Because the interference term oscillates as μ goes to 1, the limit of the Legendre representation of the residual cross section at small angles may not be well defined where the nuclear term is no longer negligible. However, if the coefficients are chosen properly, the effect of this region will be small because the Coulomb term is large.

2. "Fake cross section" approach

The "fake cross section" representation workaround has two steps:

- Start the "nuclear+interference" data tables at some finite incident energy where nuclear effects become noticeable. This eliminates the incident energy divergence in the tabulated data.
- Cut-off the "nuclear+interference" term at small angles. At small angles, Coulomb scattering dominates and must be handled in particle transport separately with techniques such as condensed history. ENDF data uses 10° as a cut-off (if remembered correctly, can't be found in documentation so far).

This process defines the "nuclear + interference" cross section and angular distribution in the CM system as:

$$\frac{d\sigma_{\mathrm{n+i}}(E)}{d\mu} = \frac{d\sigma_{\mathrm{el}}(E)}{d\mu} - \frac{d\sigma_{\mathrm{Coul}}(E)}{d\mu} \tag{46}$$

$$\sigma_{\mathrm{n+i}}(E) = \int_{\mu_{\min}}^{\mu_{\max}}\frac{d\sigma_{\mathrm{n+i}}(E)}{d\mu}d\mu \tag{47}$$

and

$$P_{\mathrm{n+i}}(\mu|E) = \begin{cases}\dfrac{d\sigma_{\mathrm{n+i}}(E)/d\mu}{\sigma_{\mathrm{n+i}}(E)} & \mu_{\min}\le\mu\le\mu_{\max}\\ 0 & \text{otherwise,}\end{cases} \tag{48}$$

Discussion point:

ENDF puts this data in MF=3 and MF=6, LAW=5. This leads to confusion since what is in MF=3 is not a partial cross section, but rather a kludge to get around the divergence. Indeed, the presence of these data in ENDF tempts one to try to heat it much like one does for neutron incident data.

Resolution:

We recommend putting these data in a special charged-particle elastic scattering `<dcrossSection_dOmega>`.

Discussion point:

When reconstructing the charged particle elastic scattering angular distributions given using the resolved resonance formats, the reconstructed data

naturally uses one of the formats mentioned here.

Requirement 52: Charged-particle elastic scattering

52.1 A `<dcrossSection_dOmega>` for charged-particle elastic scattering data
52.2 An element for "nuclear+interference" data, containing
  52.2.1 Either Legendre moments for $a_\ell(E)$ and $b_\ell(E)$ as complex and real values respectively;
  52.2.2 or complex Legendre moments for $c_\ell(E)$;
  52.2.3 or a combination of fake cross section and fake angular distribution.
52.3 A location to denote the cut-off angle (since it may not be ENDF's default 10°)

Discussion point:
The approaches described in this section can also be applied to charged particle inelastic scattering or charged particle transfer reactions.

C. Fission reactions

For application purposes, fission is regular reaction. Physically, fission is a continuum of reactions all lumped together for practicality. Thus, while it fits neatly in our top level hierarchy, at the lowest levels, there are many data types we would like to include that differ markedly from those in other reactions. That said, fission can be treated straightforwardly as a regular `<reaction>`. In this section we will elaborate on how to do this and reiterate requirements that must be met to support fission data.

Discussion point:
The top level fission reaction should just be called "fission," the name should not be overly complicated.

Discussion point:
One problem with this recommendation: we may want to break fission up into multiple reactions, for example by how many prompt neutrons are released. If we require the reaction to always be named "fission", we can't assign unique names to each of these reactions.

At the top level, the fission cross section includes not only total fission but multi-chance fission when zero or more neutrons are emitted prior to fission, based on the relative branchings for neutron emission and fission. This process is distinct from pre-equilibrium neutron emission. While the total fission cross section includes up to fourth-chance fission, no further information is available regarding reaction products from the different chances in existing ENDF evaluations. In the future, we foresee having product distributions associated with each fission chance.

Therefore, the most reasonable thing to do would be to have one `<crossSection>` element for each fission chance, each named in such a way as to make clear that they are parts of the total fission cross section (e.g., '`first_chance`'). The various chances are required to sum to the total fission cross section.

The reaction products from fission can be separated by timing into prompt, those emitted immediately after scission while the hot, excited fragments are cooling by neutron and gamma emission, and delayed, those that are emitted after processes occurring on a slower time scale such as beta decay. The prompt products include only neutrons, gammas and fission products. (The fission product yields (FPYs) are discussed in the following section.) The delayed products include not only neutrons and gamma but also electrons and neutrinos arising from the subsequent decays of the initial fission products following prompt emission. The delayed neutrons are further separated into time groups according to ranges of decay half-lives.

Discussion point:
On the subject of delayed neutrons, most time dependent group data for important (and measurable) fissionable nuclides are derived from (or adjusted to) agree with measurements (Keepin, 1957) with summation calculations being used "straight" only where such data does not exists. They can not be calculated directly from fission yields unless the FPY and decay becomes much improved than is currently available.

Currently ENDF includes an energy-dependent average energy release for fission which includes energy deposition from all the prompt and delayed emission products. The fission products are represented by post prompt neutron emission fission fragment kinetic energies. The components of the generalized energy dependent Q values include prompt neutrons and gammas as well as delayed neutrons, gammas, electrons and neutrinos. These are currently stored in a single ENDF file and, in GND, the coefficients describing the energy dependence are tabulated according to, for example, `promptNeutrons`, `delayedGammas` in the `reactionProducts` element. It may be advantageous to replicate the prompt neutron, prompt gamma, and delayed neutron contributions to the energy release within each of the `product` elements as long as the values match those at the top level. Indeed,

the energy release here should be equal to the constant pseudo-Q value for fission (MT=18 in ENDF).

Discussion point:
In ENDF, we noted that all n-th chance cross sections all had the same Q value as the total fission Q value. If this value isn't used it should be deprecated.

Discussion point:
In ENDF-6, the Q value is constant while the energy release is now energy dependent (MT=458 in ENDF). The constant Q-value should be deprecated, as mentioned above.

Both prompt and delayed neutrons are included under the `neutron` product element. Current information in ENDF includes the average neutron multiplicity $\overline{\nu}$ for prompt and delayed emission, the prompt fission neutron spectrum (PFNS), and the delayed neutron spectrum (DFNS). The delayed component includes DFNS values for different incident neutron energies separated according to time constant. Only prompt gamma emission is included under the `gamma` product element. Usually only prompt gamma energy deposition is included. However, there is room for the gamma multiplicity and energy spectra that could be made more use of.

The PFNS is unit normalized so that multiplying the PFNS at one energy by the $\overline{\nu}$ for the same energy gives the correct energy distribution. The average PFNS is typically generated by the Madland-Nix Los Alamos model and then stored in tabular form. These prompt neutron data should be augmented by the neutron multiplicity distribution, $P(\nu|E)$. Recent model calculations have shown that the shape of the PFNS depends on neutron multiplicity. For some applications, $P(E'|E,\nu)$ distributions may be useful but sampling from such distributions in lieu of an inline model of complete prompt fission events would be superior if possible at a later time. Meanwhile, the combination of $P(\nu|E)$ with the PFNS at a given E should be sufficient for most applications, particularly those involving large systems and average quantities.

Covariances on the PFNS are currently allowed in ENDF. These are relevant between incident energies, outgoing energies, and between incident and outgoing energies.

Discussion point:
Right now spontaneous fission is stored under radioactive decay. What about making a spontaneous fission reaction? Yes, I know a lot of things are different – no energy dependence etc. but it is a close cousin to neutron-induced fission, especially with respect to fission product yields and if they're moving over under a reaction called fission, why not this also? In addition, the fission reaction structure should be applicable to any other projectile inducing fission, including photofission.

Resolution:
Spontaneous fission is a decay mode of many actinides and is covered by the particle properties database discussed in section II.D and detailed in Ref. (WPEC Subgroup 38, 2015a). Spontaneous fission data would go in a `<decayProducts>` element.

Requirement 53: Fission

53.1 Allow total fission cross section to be broken out by chance in the `crossSection` element with "`1st_chance_fission`", "`2nd_chance_fission`", etc. elements. Ensure sum rules are obeyed so that the total fission cross section is retained.
53.2 Allow fission product data to be given for "`total_fission`" and "`1st_chance_fission`", "`2nd_chance_fission`", etc. to link to the products in "`total_fission`"
53.3 Allow Fission Product Yield data (see next section for a discussion)
53.4 Allow spontaneous fission data in the particle properties database
53.5 Allow breaking fission energy release into multiple components inside the reaction `<Q>`.
53.6 Allow for emission of fission fragments (products), neutrons and gammas in separate `product` elements. Electrons and neutrinos could be allowed for if users specify need but so far the tabulated deposition in the fission energy release should suffice.
53.7 Allow energy-dependent prompt, delayed and total $\overline{\nu}$ in the `<multiplicity>` element. Ensure sum rules obeyed for each energy.
53.8 Allow $P(\nu|E)$ for prompt neutrons.
53.9 Allow PFNS using tables, Madland-Nix model or other other outgoing energy distribution form supported by ENDF
53.10 Break out delayed neutrons by time group, and indicate the delayed time constant for each group. The number of time groups should not be limited to 6. Each group defines its own $\overline{\nu}$, outgoing energy spectrum and time constant.
53.11 Allow all emitted neutrons and gammas to

have incident-energy dependent multiplicities and energy-angle spectra.

53.12 Allow for energy-energy covariances in PFNS. There should be the possibility for incident energy, $E - E$; outgoing energy, $E' - E'$; and incident-outgoing energy, $E - E'$, covariances.

53.13 In addition to neutron- and γ-induced fission, also allow for fission induced by other light projectiles.

Discussion point:

Should we allow $P(E'|E, \nu)$ data for prompt neutrons? If we really want to do this, then we should only do 1 or 2 multiplicities near the average (at a given energy) since otherwise it becomes too unwieldy to try and generate decent spectra for outlying values of ν.

Discussion point:

On the subject of the 6 delayed time groups data, should we even attempt to connect the delayed data to the particle properties data and the Fission Product Yields? The 6 time groups are really effective time groups. The real process involves hundreds of individual beta decays, at least one for each independent fission product. These are averaged over in some fashion to product the time groups.

Resolution:

The decay data and fission product yield coupling is not in good enough shape to do much better than is currently done in ENDF.

D. Fission product yields

In the 2012 Working Party on Evaluation Cooperation (WPEC) meeting, two new subgroups were created: SG-37 to investigate Fission Product Yields (FPYs) and SG-38 to define a possible replacement for the ENDF-6 nuclear data format. In the May 2013 SG-37 meeting, many new theoretical and experimental results were presented and new evaluations and evaluation techniques were presented. The new evaluations provide extensive covariance data which cannot be accommodated in the ENDF-6 format. However, users require these covariance data for performing uncertainty quantification in many applications. The concurrent development of the GND format allows us to address many shortcomings of the ENDF-6 format and define a new data structure that can meet future needs of members of the SG-37 group. In this section, we summarize the results of the SG-37 and SG-38 discussions.

1. Introduction

The independent fission product yields (IFPY) are the yields of nuclear fragments following fission and prompt neutron and gamma emission. The cumulative fission product yields (CFPY) are the fragment yields after the fragments undergo further beta and other decays. Cumulative yields at two different times t_1 and t_2 are connected through a matrix:

$$\mathrm{CFPY}_i(E, t_1) = \sum_{ij} Q_{ij}(t_1, t_2)\ \mathrm{CFPY}_j(E, t_2) \quad (49)$$

The independent yield is then

$$\mathrm{IFPY}_i(E) = \mathrm{CFPY}_i(E, t = 0) \quad (50)$$

The Q matrix connects the independent yields with the cumulative yields at a finite time:

$$\mathrm{CFPY}_i(E, t) = \sum_{ij} Q_{ij}(t, 0)\ \mathrm{IFPY}_j(E) \quad (51)$$

This implies that, in practice, only IFPY *or* CFPY along with the Q-matrix may be needed. There are likely situations where both are needed, especially when fitting multiple kinds of experimental data. It is typically easier to measure the CFPY while applications typically prefer IFPY.

The Q-matrix is a sparse matrix derivable from knowledge of the fission fragment decays. There are several codes that can compute the Q-matrix from an ENDF-formatted decay sub-library. Although the Q-matrix is a derived quantity, it is derived from data potentially not associated with the FPYs tabulated (e.g., JEFF yields could in principal use ENDF/B decay data) so should be associated with the IFPY and CFPY. Also, implicit in the FPY in JEFF and ENDF/B libraries is the time-integration that sets the maximum decay time of parent nuclei of interest to the evaluator or application.

For general isotopic inventory calculations (without uncertainties) independent fission product yields are required. In ENDF/B-VIII.1, these are stored as grouped spectra "thermal", "fast" and "14 MeV i.e. DT" (Chadwick, 2011). In other cases, the cumulative fission product yields can be used to truncate decay chains to reduce the number of nuclides included in cutdown calculations.

Fission product yields are currently stored in their own sub-library in the major evaluated data libraries (e.g., ENDF/B-VII.1), but conceptually they really belong in the description of emitted particles from the fission reaction.

The uncertainties on the yields can be used to give a first order estimate of uncertainty on fission product inventories. However, to get a reasonable estimate of uncertainty on calculated inventories of fission products, the covariance terms for the combination of independent yield are required. How these quantities are going to be defined and used is something that SG-37 is still discussing. That said, it appears that we will need to store the sparse matrix of correlated independent yields (up to about 2000 by 2000, of which probably less that 10% will be non-zero).

In the process of evaluating yields, one may derive covariance data relating the yield of an isotope/isomer as a function of incident energy and covariance data relating yields from different isotopes/isomers. In addition, as the Q-matrix is derived from decay data which also have uncertainties on branching ratios, the Q-matrix may also have covariance data. The branching ratios enter into the Q-matrix linearly so the covariance calculation is straightforward. The uncertainties on half-lives is typically not so important except in the few cases of a long lived product whose half-life exceeds the integration time used to compute the Q-matrix. In this case, uncertainty propagation is very non-trivial since the half-life dependence is strongly nonlinear.

In future, models will be used to produce energy dependent yields like those prototyped in UKFY4 and GEFY. Thus energy related covariance terms may be required, perhaps combined in some way with the independent yield nuclide covariance terms. However, this will not be something evaluations will include for at least several years but it that needs noting in the format specification.

2. Existing ENDF format

The ENDF-6 format make provisions for storing the IFPY in MT=454 and CFPY in MT=459. Both FPY's use the same ENDF-6 format, namely tables of (I, YI, dYI), with I denoting the isotope/isomer in question, YI the corresponding yield and dYI the uncertainty on the yield. The yields are given for several incident energies E.

The ENDF-6 incident energy interpolation rule is specified but poorly enforced. For neutron induced fission yields, three energies are typically given which correspond to mean energies for "thermal", "fission spectrum", and "14 MeV" neutrons. In practice, the yields change slowly with incident energy so this has proven to be a problem only in a few applications.

Deuteron, alpha, photonuclear and other particle induced yields in addition to the traditional neutron and spontaneous yields were reported. The ENDF format has provisions for all of these.

The ENDF format does not provide a way to store fission yield covariances nor does it provide a way to store the Q-matrix.

3. Detailed FPY requirements

During the WPEC/SG-37 meeting, D. Brown presented some ideas on possible data structures and began a dialog with members of WPEC/SG-37. As a result of subsequent conversations, D. Brown developed a list of requirements for a new FPY data structure.

Requirement 54: Fission Product Yields (FPYs)

54.1 Clear rules for interpolation rather than a few vaguely defined groups (e.g., "thermal", "fission spectrum", "14 MeV"). Do not implicitly include spectrum averages in values.

Discussion point:

ENDF's energies really are group averages. Should the fact that they are group averages be advertised? Should we also put in the group flux somehow?

54.2 Clearly defined range of validity of evaluation that can be matched to other reaction data. This may be nothing else than the `<evaluation>`-wide `Elow` and `Ehigh`.

54.3 Clear location in the GND reaction hierarchy

54.4 Any incident particle (or none)

54.5 Per isotope/isomer yield ($Y_i(E)$), identical data structure for IFPY and CFPY

54.6 Per isotope/isomer yield uncertainty ($dY_i(E)$), identical data structure for IFPY and CFPY

54.7 Facility to store per isotope/isomer covariance on yield ($\Delta^2 Y_i(E, E')$), identical data structure for IFPY and CFPY.

54.8 Facility to store cross-isotope/isomer covariance ($\Delta^2 Y_{ii'}(E_i, E_{i'}; E_{i'}, E_{i''})$), identical data structure for IFPY and CFPY. Only IFPY's may be correlated with IFPY's and CFPY's with CFPY's, the Q-matrix couples the IFPY and CFPY.

54.9 Facility to optionally store the Q-matrix which connects the IFPY and CFPY. The Q-matrix has the following requirements:

- **54**.9.1 The upper cut-off integration time (`tStop`) used in the generation of the Q-matrix must be stored.
- **54**.9.2 The lower cut-off integration time (`tStart`) by default should be 0 s, but there should be an option to denote it as well.
- **54**.9.3 `tStart` and `tStop` could be associated with a `representation`
- **54**.9.4 The Q-matrix may be derived from IFPY

and a given CPFY *or* it may be derived from decay data. A mechanism for denoting either possibility shoudl be given.

54.10 Facility to denote which (if any) of IFPY, CFPY and or Q-matrix is a derived quantity. One simple solution would be to associate them with the appropriate 'eval' `representation`.

4. Discussion of possible implementations

Figure 38 shows an example of where fission product yields could fit in the GND-1.7 reaction hierarchy. As fission products describe the emitted particles of a fission event, it is logical to place them in the fission `<reaction>`s `<outgoingChannel>` of the corresponding `<reactionSuite>`. The collection of all fission product yield data is assembled in a `<fissionProducts>` section. The FPY section's `style` attributes specifies which of the IFPY and CFPY is the original source distribution. As spontaneous fission is a decay mode, `<fissionProducts>` sections can also appear in a particle properties database as spontaneous fission decay products.

Within the `<fissionProducts>` section, we imagine an `<independentYields>` section for IFPY, a `<cumulativeYields>` section for CFPY and possibly a `<yieldsConversionMatrix>` section to store the Q-matrix. We expect the markup for IFPY and CFPY be identical, as in the ENDF-6 format. Figures 39 and 40 show two different possible arrangements for data in the IFPY and CFPY sections.

Figure 39 shows one option. Here the yield tables use a modified version of the GND `<XYs1d>` markup (GND's `<XYs1d>` markup is equivalent to the `<interp1d>` markup used elsewhere in this document). The `<XYs1d>` markup is attractive for several reasons:

- The interpolation rule specification is well developed.
- Fudge, the main tool for manipulating GND data, has strong data structures for storing X-Y data, including linearization, plotting, etc.
- All data for one nuclide are collected together in a simple, readable way.
- By using standard data structures for each yield, we can also use standard covariance data structures for yield uncertainty/covariance data

The GND's `<XYs1d>` markup is a general markup used for data consisting of X-Y pairs. In our case, we would like to add dY's as well. The current `<XYs1d>` markup also allows for only one `<data>` tag whereas we imagine one per nuclide.

Figure 40 show another option for storing FPY. Here all data are stored in the GND `<table>` markup. This markup is quite general and compact. It can accommodate any number of isotopes simply by adding another column (or pair of columns if dY is included). We would need to add a provision for specifying an interpolation rule in energy as this is not already provided by the current `<table>` markup. With this, we would need to add quite a bit of coding to Fudge in order to generate plots and manipulate the yield data.

Discussion point:

On this option, we have to keep in mind that, in general, there are files with about 1000 FPY data for about 3 incident (neutron) energies. I would prefer option of Fig. 39. To imagine thousands of elements in a horizontal array as described in the option of Fig. 40 is a little bit impractical.

Resolution:

A 1000 x 3 table may be silly and unworkable. However that arrangement is the most ENDF-like, so we put it in as an option.

Discussion point:

About the data structure for FPY covariance data, it was thought that ENDF compact format developed and used to store large covariance matrices would be suitable for this problem. However, there is no such option proposed in this requirements document.

Resolution:

In GND and the new data structure there is agreement that there will only be one covariance matrix data structure and it will be clearer than what is in ENDF. For each dataset that has covariance data, there will be a link (with a URL) to its own covariance and any (and all) cross covariances with other datasets. It is hoped that this arrangement can be made practical for FPY's so we don't have 1000 mini-FPY tables, each with 1000 URL's pointing to 1000 mini-covariance matrices.

The Q-matrix should be stored in its own section, here called `<yieldsConversionMatrix>`. GND already provides a `<matrix>` markup and it is natural to store the Q-matrix itself here. However we need to know how each row/column maps to a yield table. To solve this, in this example we provide the URL to the data for each row/-column in the IFPY and CFPY tables. It is unclear at this time if this is the optimal way of referencing column

```
<reactionSuite projectile="n" target="U235" version="GND 1.7"
               xmlns:xlink="http://www.w3.org/1999/xlink" projectileFrame="lab">
  <styles>...</styles>
  <documentations>...</documentations>
  <aliases>...</aliases>
  <particles>...</particles>
  <resonances reconstructCrossSection="true">...</resonances>
  <reactions>
    <reaction label="0" outputChannel="n + U235" date="2020-15-   " ENDF_MT="2">...</reaction>
    <reaction label="1" outputChannel="total fission" date="2020-15-   " ENDF_MT="18">
       <crossSection>...</crossSection>
       <outputChannel genre="NBody">
        <Q>...</Q>
        <products>
          <product name="n" label="n" emissionMode="prompt">...</product>
          <fissionProducts>
            <yields style="eval">
              <independentYields>...</independentYields>
              <yieldsConversionMatrix>...</yieldsConversionMatrix></yields>
            <yields style="recon">
              <cumulativeYields>...</cumulativeYields></yields>
          </fissionProducts></products></reaction></reactions>
          ...
</reactionSuite>
```

FIG. 38 A sample GND `<reactionSuite>` demonstrating where the fission product yields could reside within a fission `<reaction>` section in the current GND format.

```
<independentYields style="eval">
   <axes>
     <axis index="1" label="energy_in"
           unit="eV"/>
     <axis index="0" label="yield" unit=""/>
     </axes>
   <yield particleID="Nd146_e0">
     <XYs1d interpolation="flat">
       <doubles>1e-05 3.45996e-13 0.0253
           3.45996e-13 ... </doubles>
       <uncertainties>...</uncertainties>
       </XYs1d></yield>
   <!-- another 779 yield elements,
        all with same axes -->
   ...
</independentYields>
```

FIG. 39 One option for storing FPY. In this variant, the yields from each isotope are given their own `<XYs1d>` section, but with common `<axes>` In this option, the yields for each fission product are given on their own grid and, owing to the re-use of the `<XYs1d>` container, whatever software is written to say plot `<XYs1d>` automatically can plot the FPY for each product. Also, as each product has its own `<XYs1d>`, the grid for one product can be refined without disrupting the other products. This arrangement also allows storing uncertainty or full covariance data directly with the yields. Cross yield covariances can be stored with the formats mentioned in section IV, in an external file.

and row elements and it depends on the way FPY's are stored in their corresponding data sections. This is illustrated in FIg. 41. In this figure, the `<grid>` elements

```
<independentYields>
  <table rows="5" columns="1561">
    <columnHeaders>
      <column index="0" name="energy" units="eV"/>
      <column index="1" name="Nd146_e0_yield"/>
      <column index="2" name="Nd146_e0_dYield"/>
          ...
        </columnHeaders>
    <data>
  <!--energy | Nd146_e0_yield | Nd146_e0_dYield |...-->
        1e-05    3.45996e-13       2.21437e-13   ...
       0.0253    3.45996e-13       2.21437e-13   ...
     500000.0    3.45996e-13       2.21437e-13   ...
     500000.0    4.01972e-13       2.57262e-13   ...
    2000000.0    4.01972e-13       2.57262e-13   ...
    2000000.0    5.452814e-09      3.489803e-09  ...
   14000000.0    5.452814e-09      3.489803e-09  ...
   20000000.0    5.452814e-09      3.489803e-09  ...
    </data></table>
</independentYields>
```

FIG. 40 An ENDF-like option for storing FPY. In this variant, all yields from all isotope are collected together in a `<table>` section. This is more compact than the other variant and has the same arrangement of information as the source ENDF file. It has the unfortunate feature that the interpolation in incident energy is lost. Also, it is not obvious how many nuclei are stored as one must parse the column names to determine that every other column is an uncertainty. Also, it is not immediately obvious how to retrofit correlation data to capture the correlations between the production of the different fragments. Finally, if one knows the yield of one product on a finer grid than the others (say because it is in the FPY peak), then it is not clear how to fit it in this grand table.

denote the linkage to the appropriate yield table to the correct column/row in the covariance matrix.

Discussion point:

Q-matrix can be computed from the decay library. Is Q-matrix something we want to store? It can be a very stringent requirement but if we computed CFPY using the Q-matrices computed from the decay data of the same library, we could store only IFPY data (and related uncertainties and correlations). In this sense CFPY can be considered as a sort of "reconstructed" FPY data as well as cross sections in the resolved resonance region are reconstructed from the resonance parameters. Obviously, this procedure would rely on a complete and consistent decay library and related uncertainties.

Resolution:

We want to allow storing Q-matrix as an option, not a requirement. Similarly, we were not requiring the evaluator to provide both the CFPY and the IFPY. However, we did want the evaluator to have the option to store either the CFPY or the IFPY and then the Q-matrix. Then the user can reconstruct what they need for their application. In the event that the evaluator has some fancy pants Bayesian scheme ;) that requires a simultaneous fit of some IFPY and some other CFPY, then that evaluator would have to store everything for the sake of internal consistency.

Discussion point:

It would be very useful to integrate a Q-matrix calculator into one or more processing/testing codes so that the IFPY and CFPY and the decay data can be brought in accord with one another.

E. Spallation reactions

Spallation reactions are reactions in which the target nucleus is struck with such force that a large number of particles are produced. For nucleon induced reactions, this typically occurs at energies $E > 100$ MeV but nearly any hadronic or leptonic projectile or gammas can cause such reactions. Spallation reactions are the main mechanism for neutron production in accelerator driven systems.

Spallation data often goes by many different names including "particle production" and "fragmentation". The main difference between "particle production" through many independent channels and "particle production" from spallation is the kinematically induced correlations between products of the reaction in the case of a spallation reaction. These correlations are quite interesting from a basic physics stand point. The incident particle deposits large amounts of energy and forward momentum into the target nucleus as the projectile stops in (or at higher energies, punches through) the target nucleus. As the energy and momentum is distributed into the target, collective motion from rescattering of the nucleons in the target build up until a plume of nucleons erupts from the target. Such reactions can provide valuable information about the nuclear equation of state and in-medium nucleon-nucleon interactions.

That said, applications involving spallation nuclear reactions do not require detailed event-by-event information. Rather, they require only the produced particle multiplicities and the energy-angle correlated outgoing probabilities. In ENDF, these data are typically stored as MT=5 reaction data in the MF=6 file.

By their nature, spallation reactions are inclusive. However, they cannot be broken easily into parts nor stored in the `<summedReactions>` portion of the hierarchy. There is no need for such a complicated mechanism for storing spallation data. We can proceed as in legacy ENDF and simply declare the spallation data to have their own `<reaction>`. The total of all the cross sections of all processes comprising the spallation reaction are placed in the `<crossSection>` element and each tabulated particle is given its own `<product>` element with variable multiplicities the full energy-angle PDFs. As the hierarchy can already accommodate these data, additional requirements are minimal:

Requirement 55: Spallation

55.1 Should allow a reaction annotation or alias. For example, "spallation".

Discussion point:

Is it enough to correlate all products under `<spallation>` or should we do something else?

Resolution:

For now, the event-by-event collective motion induced in the emitted particles via the collision kinematics is not considered in ENDF. These correlations might be important in future applications.

```
<yieldsConversionMatrix tStart="0 s" tStop="100 yr">
  <gridded2d>
    <axes>
      <grid index="2" label="row_yields" unit="" style="parameters">
        <parameter xlink:href="../../../../independentYields/yield[@particleID='Nd146_e0']"/>
        <!-- another 779 links to other yields in the independentYields -->
        ...
        </grid>
      <grid index="1" label="column_yields" unit="" style="parameters">
        <parameter xlink:href="../../../../cumulativeYields/yield[@particleID='Nd146_e0']"/>
        <!-- another 779 links to other yields in the cumulativeYields -->
        ...
        </grid>
      <axis index="0" label="Q_matrix" unit=""/></axes>
    <array shape="780,780" symmetry="lower">
      <values length="303810">...</values></array></gridded2d>
</yieldsConversionMatrix>
```

FIG. 41 An option for storing the Q-matrix. The matrix itself is stored in a `<array>` section which could be sparse or dense. The identities of the rows and columns are denoted in the `<grid>` elements with links to the appropriate yield table. The time cut off used to generate the yield conversion matrix from decay data could optionally given as an attribute in the top element.

F. Radiative capture

Here we detail needs for radiative capture data, including (n,γ), (p,γ), or any other reaction in which the only reaction product is a residual nucleus and emitted gammas. These reactions may be treated using R-matrix theory and the data stored in resonance data structures or they may be treated using other reaction models stored pointwise. If radiative capture is treated with the R-matrix formalism, often one works in the Reich-Moore approximation, but one could treat with full R-matrix using data structure consistent with requirements in this document.

Radiative capture data can be handled just like any other reaction that produces gammas, with one exception: primary gammas. Direct capture at low energies can result in the emission of a primary gamma. As the projectile's energy E increases, the energies of the primary gammas increase as well, in accordance with the equation:

$$E'_\gamma = \frac{m_{\text{targ}}}{m_{\text{proj}} + m_{\text{targ}}} E + E'_{\gamma 0} \tag{52}$$

where m_{targ} and m_{proj} are respectively the masses of the target and projectile, and $E'_{\gamma 0}$ is the energy of the gamma for a projectile with zero energy. As the neutron moves up in incident energy, the primary gamma's energy appears to move with it.

In ENDF, primary gammas are a mess, with two separate implementations (MF=12 and MF=6). Both ENDF approaches are kludges The primary gammas and accompanying cascade gammas are unique to an isotope and are a potential tool for isotope identification through active interrogation.

Requirement 56: Primary gammas

56.1 There shall be a special markup to denote that a gamma is a primary gamma.

G. Thermal scattering law

1. Introduction

In the 2015 Working Party on Evaluation Cooperation (WPEC) meeting, Subgroup 42 "Thermal Scattering Kernel S(α,β): Measurement, Evaluation and Application" was formed. At the kick-off meeting for this Subgroup, several issues arose:

- The need for directional information from thermal neutron scattering events for moderator design in new spallation neutron sources;
- New approaches to the evaluation of thermal neutron scattering data;
- New approaches to uncertainty propagation for thermal neutron scattering data; and
- New experimental activities seeking to measure thermal neutron scattering cross sections on technologically important materials.

All of these require improvements to the legacy ENDF format. Fortunately, several members of Subgroup 38 were present at this meeting and were able to recruit several of the Subgroup 42 participants to help draft the requirements for thermal scattering data in the new nuclear data structure.

Thermal neutron scattering law (TSL) data apply to the situation where the de Broglie wavelength of an incident neutron is on the order of interatomic distances and the neutron's energy is on the order of the excitation (vibration, rotation, etc.) energy of the medium in which it interacts. The structure and dynamics of the scattering medium determine the peculiarities of neutron scattering at low incident neutron energies ($E < 1 - 10$ eV). Thermal neutron scattering is typically formulated using the theory of Van Hove (VanHove, 1954) which we detail here following the treatments in Refs. (Hehr, 2010) and (Cacuci, 2010).

TSL data are given in sub-library 12 (NSUB = 12) in the ENDF6 format (using MF=7, MT=2 and MF=7, MT=4 data structures). This sub-library provides dimensionless scattering kernels on a grid of dimensionless momentum and energy transfer to describe thermal neutron scattering. The sub-library is organized by a nuclide (scatterer) in a given material. For example, in the ENDF/BV-II.1 TSL sub-library, we have data for Be in beryllium oxide, O in beryllium oxide, C in Graphite, etc. In some cases, only the most important scatterer in a material has the evaluation. For example, we have `HinH2O`, or hydrogen in light water, but there is no evaluation for `OinH2O`, implying that usage of the free gas model for thermal neutron scattering by oxygen in light water is an acceptable approximation. Some evaluations have the data at one temperature: for example, data for thermal neutron scattering by H in liquid parahydrogen (H_2 with both atomic spins aligned so $I = 0$) are given at $T = 20.0°$K only. However, many evaluations are given for a number of temperatures T. For example, $S(\alpha, \beta; T)$ data for `UinUO2` (U in uranium dioxide) are given at eight different temperatures.

Discussion point:

The materials in the current ENDF libraries could be organized using the `<metaEvaluation>` markup in section III.B.

When using, for example, ENDF/B-VII.1 TSL data, it is expected that nuclear data processing codes can read $S(\alpha, \beta, T)$ data and generate differential cross sections, $d^2\sigma(E, T)/dE'd\Omega$, as well as the integral data (such as, integral cross sections $\sigma(E, T)$, average scattering cosine $\bar{\mu}(E, T)$, average E', etc.) in proper physical units (barn per eV per sr, barn, eV, etc.) for incident neutron energies E and neutron scattering with the energy E' and scattering cosine μ.

Discussion point:

It was felt at the May 2014 Paris meeting, that we should consider focusing on storing only legacy ENDF data and possibly the phonon spectrum $\rho(\omega)$ and defer all subsequent discussions until the formation of a special WPEC subgroup that can specifically deal with TSL issues. In May 2015, SubGroup 42 "Thermal Scattering Kernel $S(\alpha, \beta)$: Measurement, Evaluation and Application" was formed.

2. Theoretical background

Working in the first Born approximation and applying a Fermi (nuclear) pseudo potential, the double differential thermal neutron scattering cross section off one class of N scatterers j with scattering lengths b_j is

$$\frac{d^2\sigma(E)}{d\Omega dE'} = \frac{k'}{2\pi\hbar k}\sum_{j,j'} b_j b_{j'} \times \int_{-\infty}^{\infty} dt e^{-i\omega t}\left\langle e^{-i\vec{\kappa}\cdot\vec{R}_{j'}(0)} e^{-i\vec{\kappa}\cdot\vec{R}_j(t)}\right\rangle \tag{53}$$

This expression includes both elastic and inelastic scattering. Here $\vec{R}_j(t)$ is the position Heisenberg operator of the j^{th} scatterer at time t, and k, and k' are the incident and outgoing neutron wave numbers. The momentum and energy transferred between the scattering neutron and the collections of scatterers are $\vec{\kappa} = \vec{k} - \vec{k'}$ and $\omega = E' - E$. The implicit dependence on the material temperature is suppressed.

In the limit of a large number of scatterers we may make the replacement $b_j b_{j'} \rightarrow \langle b_j b_{j'}\rangle$ and further, assuming no correlation between the scattering lengths of different nuclei,

$$\begin{aligned} \langle b_j b_{j'}\rangle &= \langle b_j\rangle\langle b_{j'}\rangle = \langle b\rangle^2 && \text{if } j' \neq j \\ \langle b_j b_j\rangle &= \langle b^2\rangle && \text{for } j = j' \end{aligned} \tag{54}$$

Using this, we define the coherent scattering cross section as $\sigma_{coh} = 4\pi\langle b\rangle^2$ and the incoherent as $\sigma_{inc} = 4\pi(\langle b^2\rangle - \langle b\rangle^2)$. Now, define an intermediate function

$$I(\vec{\kappa}, t) = \frac{1}{N}\sum_{j,j'}\left\langle e^{-i\vec{\kappa}\cdot\vec{R}_{j'}(0)} e^{-i\vec{\kappa}\cdot\vec{R}_j(t)}\right\rangle \tag{55}$$

The intermediate function can be computed from condensed matter theory assuming that the dynamics (described in terms of vibrational eigenmodes or phonon-type spectra) and structure (e.g., a certain order or correlations in the positions of scatterers in space) of the medium of interest are well understood.

In terms of the intermediate function, the scattering kernel is

$$S(\vec{\kappa}, \omega) = \frac{1}{2\pi\hbar}\int_{-\infty}^{\infty} dt e^{-i\omega t} I(\vec{\kappa}, t) \tag{56}$$

Both $S(\vec{\kappa}, \omega)$ and $I(\vec{\kappa}, t)$ may include all j, j' in the sum in Eq. (53) or they may be broken out into $j = j'$ (self) and

$j \neq j'$ (distinct) contributions. This is useful because the incoherent cross section only contains the self-correlation between an atom at time $t = 0$. We have

$$\frac{d^2\sigma(E)}{d\Omega dE'} = \frac{k'}{k}\left[\sigma_{coh} S(\vec{\kappa},\omega) + \sigma_{inc} S_s(\vec{\kappa},\omega)\right] \quad (57)$$

where

$$S(\vec{\kappa},\omega) = S_s(\vec{\kappa},\omega) + S_d(\vec{\kappa},\omega) \quad (58)$$

Discussion point:

Reaction annotations could be used to split the scattering kernel into "self" and "distinct" parts. This is useful for a model-based evaluation where both components can be computed separately.

In practice, one assumes that we may average over orientation of the scatterers such that we can replace the directional dependence of $\vec{\kappa}$ with a directionless κ dependence. Also, one uses the scattering kernel rewritten in terms of the dimensionless variables α and β so

$$\mathcal{S}(\alpha,\beta) = \frac{k_B T}{\hbar} S(\kappa,\omega) \quad (59)$$

where $\alpha = (E' + E - 2m_n\sqrt{EE'})/Ak_BT$ and $\beta = (E' - E)/k_BT$.

Discussion point:

Whether we use α and β or κ and ω, we have reduced the parametric dependence of the scattering kernel to three. These are κ, ω and an implicit material temperature dependence. This makes storing the scattering kernel directly in ENDF feasible.

Resolution:

We should stick to storing the kernel in terms of α and β for backwards compatibility

Discussion point:

New experiments from NCSU/RPI/ORNL collaboration will directly measure the $d^2\sigma(E,T)/dE'd\Omega$. This is equivalent to measuring the full scattering kernel. Storing the covariance on the full scattering kernel may be unfeasible. Storing the covariance on data using the approximations and distinctions below may be feasible.

Discussion point:

ENDF thermal scattering data can have a large dynamic range. To accommodate this, ENDF manual recommends the following:

> For down-scattering events with large energy losses and for low temperatures, β can be large and negative. The main contribution to the cross section comes from the region near $\alpha + \beta = 0$. Computer precision can become a real problem in these cases. As an example, for water at room temperature, calculations using equation (7.6) for incident neutrons at 4 eV require working with products like $e^{80} \times 10^{-34}$. For liquid hydrogen at 20 Kelvin and for 1 eV transfers, the products can be $e^{300} \times 10^{-130}$. These very large and small numbers are difficult to handle on most computers, especially 32-bit machines. The LLN flag is provided for such cases: the evaluator simply stores $\ln(S)$ instead of S and changes the interpolation scheme accordingly (that is, the normal log-log law changes to log-lin). Values of $S = 0.0$ like those found in the existing ENDF/B-III thermal files really stand for some very small number less than 10^{-32} and should be changed to some large negative value, such as -999.

The fact that we are not limited by ENDF's precision means that we should avoid this problem.

The TSL data can be simplified further by taking advantage of the elastic limit ($\omega \to 0$) or by making several approximations including

- The incoherent approximation, where the "distinct" part of the scattering law is neglected (i.e., setting $S_d = 0$ above).
- The Gaussian approximation, where the "self" part of the intermediate scattering function may be written in terms of the material's phonon frequency $\rho(\omega)$.
- The short collision time approximation

Requirement 57: Thermal scattering kernel

57.1 Allow TSL data to be broken out into separate reactions as specified by the evaluator. Each reaction is treated independently for the purposes of neutron transport.

57.2 Denote the energy range for which these data are used. The $E_{max} = 5$ eV limit in ENDF is convention and has no general physical justification.

57.3 Allow reactions to be annotated by combinations of `self`, `distinct`,

`coherent`, `incoherent`, `tsl_elastic` and `tsl_inelastic` labels. Because of this flexibility, care will need to be taken by evaluators to ensure that double counting does not occur.

57.4 All reaction data contained in `<dcrossSection_dOmega_dE>` or `<dcrossSection_dOmega>` elements, depending on the evaluators needs.

57.5 Provide a mechanism by which multiple scatterers within one material may be associated (or better yet, collected) in one file. This way the connection between say `BeinBeO` and `OinBeO` is made explicit.

57.6 If the reaction data are be broken out by scatterer (e.g., `HinH2O`), the stoichiometric fraction of each class of scatterer must be given.

57.7 The scattering kernel $\mathcal{S}(\alpha, \beta, T)$ can be given as an interpolation table or using one of the approximations or distinctions given below in subsections V.G.4-V.G.6

57.8 Coherent or incoherent cross sections are associated with their respective scattering kernels.

Discussion point:

Annotations might also be used to denote "`tsl_elastic`" and "`tsl_inelastic`" data as TSL data does not have the same two-body kinematics of higher energy data. When $\omega \to 0$, $E = E'$ in the lab frame and the center of mass frame is meaningless.

3. Gaussian approximation of the self part of the scattering kernel

By making the so-called Gaussian approximation to the intermediate scattering function (Hehr, 2010; VanHove, 1954) the self part of the scattering kernel can be written in terms of the material phonon spectrum $\rho(\omega)$:

$$\mathcal{S}_s(\alpha, \beta) = \frac{1}{2\pi} \int_{-\infty}^{\infty} dt\, e^{i\beta t} e^{-\gamma(t)} \tag{60}$$

where

$$\gamma(t) = \alpha \int_{-\infty}^{\infty} d\omega \rho(\omega) \left(1 - e^{-i\omega t}\right) \frac{e^{-\omega/2}}{2\omega \sinh(\omega/2)} \tag{61}$$

This Fourier transform is coded in R.E. MacFarlane's LEAPR module of NJOY (MacFarlane, 2012) and can be used to generate the inelastic scattering kernel.

We note that if the full scattering kernel is well approximated by only the self term and in this Gaussian approximation, the entire scattering kernel can be specified with the phonon spectrum. This spectrum may have a discrete portion and/or a continuous portion. Nevertheless, this offers us a compact way to encapsulate the scattering kernel and it provides us with a two-dimensional object that we can specify covariance on.

Requirement 58: Gaussian self-part scattering kernel

58.1 An encapsulating element that specifies that these data is the self part of the scattering kernel only.

58.2 The phonon spectrum $\rho(\omega)$ as a discrete and/or continuous distribution.

58.3 Optionally, a `link` to the covariance on the phonon spectrum.

Following (Cacuci, 2010), we can expand the time-dependent part of the scattering kernel to arrive at the phonon expansion:

$$e^{-\gamma(t)} = -\alpha\lambda_s \sum_{n=0}^{\infty} \frac{\alpha^n}{n!} \left[\int_{-\infty}^{\infty} d\omega P_s(\omega) e^{-\omega/2} e^{-i\omega t}\right]^n \tag{62}$$

where $P_s(\omega) = \rho(\omega)/\left[2\omega \sinh\left(\omega/2k_BT\right)\right]$ and the Debye-Waller coefficient is

$$\lambda_s = \int_{-\infty}^{\infty} d\omega P_s(\omega) e^{-\omega/2} \tag{63}$$

The n^{th} term in this expansion is identified with the number of phonons involved in the collision (Cacuci, 2010). This expansion allows us to arrive at two approximations given in the ENDF format manual that must be grandfathered into the new data structure: elastic coherent, elastic incoherent and the short collision time approximations. In the case of the elastic (in)coherent data, we set $E = E'$ which then forces $\beta \to 0$ and simplifies the phonon expansion to the first term, the zero phonon limit. Alternatively, in the limit of large n, we arrive at the short collision time approximation (called this because the large number of collisions implies a short time for each individual collision).

4. Coherent elastic scattering

For crystals (polycrystalline materials), the information about the crystal structure is expressed in terms of the so-called Bragg edges (a discrete set of energies $E_j \sim 1$ meV - 1 eV) and a set of crystallographic structure factors s_j associated with E_j and a neutron scatterer in a crystal unit cell. In addition, one has to estimate the temperature dependent Debye-Waller coefficient W' (in the units of eV^{-1}). Then it is possible to generate the

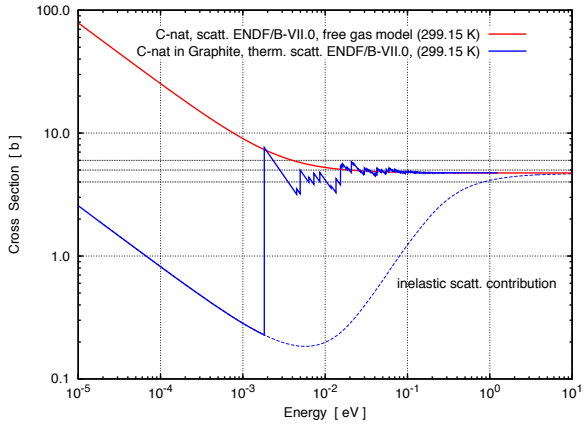


FIG. 42 Elastic scattering cross-sections of carbon at room temperature (free gas model) vs. thermal scattering cross-sections of carbon in graphite at room temperature.

data structure that can be used to generate the contribution of coherent elastic neutron scattering into the thermal neutron scattering kernel, scattering cross sections, etc., for a given scatterer in the polycrystal.

Figure 42 compares the different elastic scattering prescriptions for two different forms of carbon. Here one can clearly see the Bragg edges in the elastic cross section.

In the early 1990's, parameterized coherent and incoherent elastic scattering were added to ENDF format. Neutrons can only elastically scatter coherently off of regular substances such as crystals. The differential cross section for such scattering can be written (Trkov, 2009)

$$\frac{d^2\sigma}{dE'\,d\Omega}(E,T) = \frac{1}{2\pi E}\sum_{i=1}^{E_i<E} s_i(T)\;\delta(\mu-\mu_i)\;\delta(E-E') \tag{64}$$

where:

$$\mu_i = 1 - \frac{2E_i}{E} \tag{65}$$

The quantity actually given in the file is $s_i(T) = S(E_i, T)$ which the ENDF manual states is conveniently represented as a stairstep function with breaks at the Bragg edges E_i using histogram interpolation. Here, we must store the structure factor $s_i(T) = S(E_i, T)$ tables in ENDF's MF=7 (note these factors are given as a histogram in ENDF, hence the notation above).

Alternatively, this cross section can be written (Cacuci, 2010):

$$\frac{d^2\sigma}{dE'\,d\Omega}(E,T) = \frac{\sigma_{coh}}{E}\sum_{i=1}^{E_i<E} f_i e^{-4WE_i}\delta(\mu-\mu_i)\;\delta(E-E') \tag{66}$$

The f_i are material dependent and related to the crystallographic structure factors.

Requirement 59: Coherent Elastic Scattering

59.1 An elastic channel reaction designator that includes the annotations `tsl_elastic` and `coherent`.

59.2 `<dcrossSection_dOmega>` element containing this data.

59.3 An list of `<interp2d>` elements containing the structure factor $S(E,T)$ in the `<distribution>` element. The ENDF manual requires the interpolation in E to be a histogram and it is unclear whether there is a need to relax this requirement. The Bragg edges are the histogram boundaries. With this requirement, $S(E,T) = S(E_i,T) \equiv s_i(T)/E$

59.4 Alternatively, specify the Bragg edges E_i and factors f_i in a `<table>` element. Must provide the coherent cross section σ_{coh} and the Debye-Waller integral W as well.

59.5 Optional `link` to `<covariance>` data on the specified parameters.

59.6 These data must be given in the lab frame as the center of mass frame is meaningless for TSL data.

5. Incoherent elastic scattering

For partially ordered systems, incoherent elastic scattering is described by

$$\frac{d^2\sigma}{dE'\,d\Omega}(E,T) = \frac{\sigma_b}{4\pi}\,e^{-2EW'(T)(1-\mu)}\;\delta(E-E') \tag{67}$$

where:

σ_b is the characteristic bound cross section (barns),

W' is the DebyeWaller integral divided by the atomic mass (ENDF data are given in eV^{-1}),

and all the other symbols have their previous meanings. The integrated cross section is easily obtained:

$$\sigma(E) = \frac{\sigma_b}{2}\left(\frac{1-e^{-4EW'}}{2EW'}\right) \tag{68}$$

Note that the limit of σ for small E is σ_b.

Requirement 60: Incoherent Elastic Scattering

60.1 An elastic channel reaction designator that includes the annotations `tsl_elastic` and `incoherent`.

60.2 `<dcrossSection_dOmega>` element containing these data.

60.3 An `<interp1d>` element collecting W' the DebyeWaller integral divided by the atomic mass

as a function of temperature.
60.4 The bound cross section σ_b, with units.
60.5 Optional `link` to `<covariance>` data on W'.
60.6 Optional `link` to `<covariance>` data of σ_b. This is a 1×1 matrix, but could be correlated with W''s covariance.
60.7 These data must be given in the lab frame as the center of mass frame is meaningless for TSL data.

6. Incoherent inelastic scattering in the short collision time approximation

In the short collision time limit, we have (Trkov, 2009)

$$\mathcal{S}(\alpha,\beta) = \frac{\exp\left[-\frac{(\alpha-|\beta|)^2 T}{4\alpha T_{\rm eff}(T)} - \frac{|\beta|}{2}\right]}{\sqrt{4\pi\alpha\frac{T_{\rm eff}(T)}{T}}} \tag{69}$$

Requirement 61: Short Collision Time Approximation

61.1 Something to denote that the short collision time approximation is used.
61.2 The effective temperature $T_{\rm eff}(T)$

Discussion point:
Should the Short Collision Time approximation be deprecated as a format? It is not used in current libraries and the approximation is only used during processing in certain cases.

7. Discussion of TSL covariance

There are two general applications possible for thermal scattering covariance data. The first is for use in nuclear system simulation codes to assess uncertainties in calculated system response parameters as a function of supplied nuclear data uncertainties. The second is to allow statistical comparison of different thermal scattering law (TSL) and cross section evaluations and measurements.

For uncertainty quantification in nuclear systems, there are not yet any simulation codes capable of utilizing differential or double-differential covariances. Any simulation code incorporating inelastic thermal scattering data will either utilize the $S(\alpha,\beta)$ TSL directly or have its own thermal library format with integral cross section and scattered angle/energy information. For the latter case, which is more common, the library is typically produced by some method of numerical integration over $S(\alpha,\beta)$. In this case, regardless of how this $S(\alpha,\beta)$ was generated, the user may propagate a supplied covariance matrix for the $S(\alpha,\beta)$ data to determine covariances for any set of differential and integral cross section data, for any library format, for any energy/angle grid structure, and without the need for evaluator-supplied sensitivity matrices. The user may also determine the resolution of differential and integral covariance data required and define group cross sections for this covariance data based on their application needs. This is all a strong argument for developing a format to provide covariances on $S(\alpha,\beta)$ directly.

For interlibrary or evaluation to experiment comparisons, we want the covariance format to be compatible with both experimental and theoretical thermal scattering data. For the latter, we want to include evaluations in the incoherent approximation based on an excitation mode energy spectrum (since this applies to all currently published TSL evaluations) as well as evaluations generated with modern methods using molecular dynamics, the time-dependent pair correlation function $G(r,t)$, or otherwise explicitly accounting for coherent interference effects in inelastic scattering. A format providing covariances on $S(\alpha,\beta)$ directly would satisfy all of this. As with uncertainty quantification, this common format would allow the propagation of the $S(\alpha,\beta)$ covariance matrix to generate covariances for any set of differential and integral cross sections, regardless of how the $S(\alpha,\beta)$ data was generated.

The dimensionality of the $S(\alpha,\beta)$ TSL is a problem. Strictly speaking, the TSL is a function of alpha, beta, sigma, and T. We can probably safely assume that the uncertainty in the nuclear scattering cross section (sigma) is small compared to uncertainties in $S(\alpha,\beta)$. We can certainly assume this for nuclear mass uncertainties. There are no uncertainties in T if we assume the TSL to be defined for exact T. However, if we include T as a dimensional parameter of the TSL covariance matrix, the matrix is tabulated as a function of six dimensions $(\alpha,\alpha',\beta,\beta',T,T')$. Even with only 10 elements per dimension, this results in a covariance matrix with 10^6 entries. An important question here is, "How important is cross-temperature covariance data?" For uncertainty quantification, the answer may be model dependent. For the model comparison, it is probably not important. There is also the question of cross-nuclide covariances, (e.g., covariances between the TSLs for Zr in ZrH_x and H in ZrH_x). We already ignore many cross-library covariances for high-energy reactions and materials out of practicality. Although it may be clear what kind of data we would like to have, if we are to have any usable thermal scattering covariance data at all, issues of practical application must be considered.

In terms of the α and β dimensions, covariances could be tabulated over coarse α and β grids without affecting the resolution of the $S(\alpha,\beta)$ data itself. $S(\alpha,\beta)$ data has a very large dynamic range that can span many orders of magnitude. Physically, very-low magnitude $S(\alpha,\beta)$ is as-

sociated with very-low probability scattering events. Although tabulating these quantities may be important for accurate convergence of integral cross sections and capturing low-probability angle/energy changes, uncertainties in integral cross sections will be very insensitive to uncertainties in very-low magnitude $S(\alpha, \beta)$. Moreover, for α and β blocks over which the variation in $S(\alpha, \beta)$ is small, retaining a fine covariance structure may be unnecessary. In summary, the α and β resolution of $S(\alpha, \beta)$ covariance data required to generate integral uncertainties of sufficient fidelity is likely much less than the α and β resolution of $S(\alpha, \beta)$ required to generate differential and integral cross sections of sufficient fidelity.

VI. DERIVED DATA FOR APPLICATIONS

One of the goals for the new structure is to be able to store processed or derived data for specific applications (see requirement **2**.2). Here we attempt to list some of the more important cases and their requirements, but we realize that our list is far from complete.

Processed data are a representation of the data in a form required by transport (or other application) codes. This kind of data is often needed for inter comparison between labs. However, each institution generally processes evaluated data into a form mainly acceptable by applications managed by that institution, and these forms depend on each institution's unique needs.

The elements and requirements we describe below are located in a few distinct places in the data hierarchy:

`<reaction>`: To store evaluated and any processed/derived data specific to a reaction

`<productionReactions>`: To store production data such as production cross sections and yields.

`<applicationData>`: To store other derived data not tied to a specific `<reaction>` but needed for transport, such as total transfer matrices. This section is not to be confused with `<representations>` contents which define things like group structures and fluxes needed in the generation of processed data.

We comment that there are cases even in ENDF where there is incomplete information known (e.g., for fission we may only know the total emitted energy for some of the minor actinides) so these "derived" or "processed" data may be all we know. Therefore, in any given data set, the `<representation>` determines what is original and what isn't.

A. General transport data

1. Product average kinetic energy and forward momentum

A product's average kinetic energy $\overline{E}'(E)$ is defined as

$$\overline{E}'(E) = \int_0^\infty dE' \int_{-1}^1 d\mu \; E' \, P(\mu, E'|E) \tag{70}$$

It is convenient to store this quantity in a linear tabulated form as it is used to check energy balance.

A product's average forward momentum $\overline{\mathrm{p}}'_{\mathrm{f}}(E)$ is defined as

$$\overline{\mathrm{p}}'_{\mathrm{f}}(E) = \int_0^\infty dE' \int_{-1}^1 d\mu \; \mu \, \mathrm{p}'(E') \, P(\mu, E'|E) \tag{71}$$

where $\mathrm{p}'(E')$ is the magnitude of the product's momentum as a function of its kinetic energy E' and the μ factor in the equation yields the projection of the product's momentum along the direction of the projectile. It is also convenient to store this quantity in a linear tabulated form.

Requirement 62: Average kinetic energy and forward momentum

- **62**.1 Shall store the projectile's averaged kinetic energy and momentum within the appropriate `<reaction>`
- **62**.2 Shall store the products' averaged projectile kinetic energy and momentum within the appropriate `<reaction>`'s `<product>` element
- **62**.3 The averaged projectile kinetic energy and momentum shall be stored as `<interp1d>` tables

2. $\bar{\mu}_{lab}(E)$

In ENDF, the average forward scattering angle in the lab frame of a reaction product, $\bar{\mu}_{lab}(E) = \int d\mu_{lab} \mu_{lab} P(\mu_{lab}|E)$ is given in the MT=251 file. Although one might view $\bar{\mu}_{lab}(E)$ as a derived quantity, it can be measured experimentally and so may have covariance data associated with it. We note that the data typically given in evaluated files are in the center of mass frame, so care must be taken when performing the frame change.

Requirement 63: Mubar

- **63**.1 Shall store the projectile's averaged forward scattering angle within the appropriate `<reaction>`
- **63**.2 Shall store the products' averaged forward scattering angle within the appropriate `<reaction>`'s `<product>` element
- **63**.3 The averaged forward scattering angle shall be stored as `<interp1d>` tables

3. CDF's from PDF's

Given a probability density function PDF(E), one can define the cumulative distribution functions (CDF) as

$$\mathrm{CDF}(E) = \int_{-\infty}^{E} dE' \mathrm{PDF}(E') \tag{72}$$

Monte Carlo codes use these cumulative distribution functions (CDF)'s to convert uniformly sampled random numbers, x, on the interval $x \in [0, 1]$ to samples consistent with the underlying PDF through

$$E = \mathrm{CDF}^{-1}(x) \tag{73}$$

For Monte Carlo transport, it is convenient to precompute the cumulative distribution functions (CDF) for each PDF. However, this is a fast calculation and precalculation is not essential, but may be advantageous. Therefore, we require a markup for CDF's as derived data to be located in the same enclosing element as its PDF.

Requirement 64: CDFs

64.1 A markup for CDF's as derived data, located in the same enclosing element as its PDF.

4. Probability tables in the URR

The unresolved resonance region parameters represent the average behaviors of resonances that cannot be resolved experimentally. Using techniques such as implemented in the PURR module of NJOY (MacFarlane, 2012) or the PURM module of AMPX (Dunn, 2002), one can convert these average parameters into the probability for a particular reaction cross section σ_x as a function of incident energy, $P(\sigma|E)$.

Requirement 65: URR probability tables

65.1 The URR probability tables shall be stored in the appropriate `<reaction>`

65.2 The URR probability tables shall be stored as `<interp2d>` objects

Discussion point:

The GRUCON code (GRUCON, 2014) can compute the conditional probabilities $P(\sigma_i|\sigma_j, E)$, properly accounting for the correlations in the probabilities of all the reaction cross sections. If we are to support these, they naturally would go in the evaluation-wide `<applicationData>`.

B. Grouped transport data

The largest use of data in ENDF format is modeling particle transport (usually neutrons, but users are sometime interested in transporting charged particles including electrons). To model the transport of particles such as neutrons one uses codes that solve the Boltzmann transport equation. The Boltzmann equation can be solved a variety of ways including with Monte-Carlo techniques, by discretizing it and solving the resulting matrix equation (also known as deterministic transport) or rarely by using the method of characteristics.

In a simplified 1-dimensional form with one target species and only elastic scattering, the Boltzmann equation is

$$\frac{1}{v}\frac{\partial f(E)}{\partial t} + \frac{\partial f(E)}{\partial x} + \rho\,\sigma(E)\,f(E) = \int dE'\,\rho\,\sigma(E')\,P(E'\to E)\,f(E') \quad (74)$$

where E is the kinetic energy of the projectile, $f(E) = f(x, E, t)$ is the flux of the particle being transported (i.e., the projectile), ρ is the target density, $\sigma(E)$ is the cross section between the projectile and the target and $P(E' \to E)$ is the probability density function (PDF) for scattering from energy E' to E. In this simplified form, the angular dependence is ignored.

For both Monte Carlo and deterministic data, it is convenient to convert cross section resonance data to a linear tabulated form and pre-heat the tabulated cross sections to pre-defined temperatures as these are computationally intensive calculations.

For deterministic transport the energy variable E must be judiciously discretized by flux-weight averaging elements of the Boltzmann equation over energy bins (also called groups). For example, the cross section becomes for group i ($E_i < E < E_{i+1}$),

$$\sigma(E) \Rightarrow \sigma_i = \frac{\int_{E_i}^{E_{i+1}} dE\,\sigma(E)\Phi(E)}{\int_{E_i}^{E_{i+1}} dE\,\Phi(E)} \quad (75)$$

and the $\sigma(E')\,P(E' \to E)$ factor in the right-hand-side of Eq. 74 becomes,

$$\sigma(E)\,P(E'\to E) \Rightarrow \mathrm{TM}_{io} = \frac{\int_{E'_o}^{E'_{o+1}} dE' \int_{E_i}^{E_{i+1}} dE\,\sigma(E')P(E'\to E)\,\Phi(E')}{\int_{E'_o}^{E'_{o+1}} dE'\,\Phi(E')} \quad (76)$$

where TM_{io} (called the transfer matrix) expresses the fact that a projectile in energy group o produces outgoing particles into energy group i. Equation 76 is a simplified version; a more accurate TM_{io} includes Legendre expansion, a multiplicity for the number of outgoing particles $M(E')$ and a conservation factor $W(E')$. The conservation factor is either 1 or E', depending on whether the number or average energy of the outgoing particles is to be conserved for each group.

Ideally, the flux $\Phi(E)$ that we weight with should be the same flux that results from solving the transport equation, $f(E)$. Clearly this presents a "chicken and egg" problem since we can't weight with a flux we haven't yet computed. Therefore, in practice various techniques have been developed to make judicious choices for the weighting fluxes. Bondarenko (see VI.B.5) and multi-band treatments are some of the more popular choices in the application community.

The more accurate representation of the transfer matrix is,

$$\mathrm{TM}_{io,l} = \frac{\int_{-1}^{1} d\mu P_l(\mu) \int_{E'_o}^{E'_{o+1}} dE' \int_{E_i}^{E_{i+1}} dE\, M(E')W(E')\sigma(E')P(E' \to E, \mu)\, \Phi(E')}{\int_{E'_o}^{E'_{o+1}} dE'\, \Phi(E')} \tag{77}$$

The grouped deterministic transport representation of Eq. 74 with Legendre notation (although still simplified) is,

$$\frac{1}{v_i}\frac{\partial f_{i,l}}{\partial t} + \frac{\partial f_{i,l}}{\partial x} + \rho\, \sigma_i\, f_{i,l} = \rho \sum_o \mathrm{TM}_{io,l}\, f_o \tag{78}$$

In addition to σ_i and $\mathrm{TM}_{io,l}$, it is convenient to group other quantities; namely, product multiplicity, energy dependent Q data (i.e., $Q(E)$), projectile kinetic energy and momentum, product average kinetic energy and forward momentum, and inverse speed (i.e., $1/v_i$). It is also convenient to have several of the quantities in linear tabulated form; namely, projectile momentum, and product kinetic energy and forward momentum.

All of these quantities can be classified by their dimension and location with the structure as,

Requirement 66: Grouped transport data

- **66**.1 With the exception of $\mathrm{TM}_{io,l}$, all grouped quantities can be stored as a list (i.e., an array of dimension 1) while $\mathrm{TM}_{io,l}$ requires an array of dimension 3.
- **66**.2 The following describes where each element shall be stored.
  - **66**.2.1 The inverse speed shall be stored within the `<evaluation>` element.
  - **66**.2.2 The cross section, energy dependent Q data, and projectile kinetic energy and momentum shall be stored within the appropriate `<reaction>` element.
  - **66**.2.3 The transfer matrix, multiplicity, and product average kinetic energy, forward momentum and average forward scattering angle shall be stored within the appropriate `<product>` element.
- **66**.3 The group structure and flux weights are stored in the `<representations>` elements for the evaluation
- **66**.4 Grouped data must clearly state the flux weighting and group structure used to derive the data as well as the original data that was grouped.

In addition, both the number and energy conserving $\mathrm{TM}_{io,l}$, should be stored as separate elements (with unique names), as both may be needed.

In the following sections, the cross section weighted average $\langle A \rangle_i$ of a quantity $A(E)$ is defined as,

$$\langle A \rangle_i = \frac{\int_{E_i}^{E_{i+1}} dE\, A(E)\, \sigma(E)\, \Phi(E)}{\int_{E_o}^{E_{o+1}} dE\, \Phi(E)} \tag{79}$$

Note, we may want to define it as

$$\langle A \rangle_i = \frac{\int_{E_i}^{E_{i+1}} dE\, A(E)\, \sigma(E)\, \Phi(E)}{\int_{E_o}^{E_{o+1}} dE\, \sigma(E)\, \Phi(E)} \tag{80}$$

$$= \frac{\int_{E_i}^{E_{i+1}} dE\, A(E)\, \sigma(E)\, \Phi(E)}{\sigma_i} \tag{81}$$

1. Inverse speed

The cross section weighted average inverse speed is defined as $\langle 1/v \rangle_i$ where $v(E)$ is the projectile's velocity as a function of its energy.

2. Multiplicity

The cross section weighted average multiplicity is defined as $\langle m \rangle_i$ where $m(E)$ is the product's multiplicity as a function of its energy.

3. Q-value

The cross section weighted average Q-value is defined as $\langle Q \rangle_i$ where Q is a reaction's Q-value as a function of its energy.

4. Projectile kinetic energy and momentum

The cross section weighted average projectile kinetic energy is defined as $\langle E \rangle_i$.

The projectile's momentum as a function of its kinetic energy E is written as $\mathrm{p}(E)$ and it is convenient to store it in a linear tabulated form. The cross section weighted average momentum is defined as $\langle \mathrm{p} \rangle_i$.

5. Bondarenko factors

The Bondarenko method represents self-shielded cross sections in terms of temperature and a background cross

section parameter σ_0 which indicates the degree of self-shielding:

$$\tilde{\sigma}^j_{x,i}(E,T,\sigma_0) = \frac{\int_{E_i}^{E_{i+1}} dE\ \sigma^j_x(E,T)\Phi(E,T,\sigma_0)}{\int_{E_i}^{E_{i+1}} dE\ \Phi(E,T,\sigma_0)} \tag{82}$$

where $\tilde{\sigma}^j_{x,i}(E,T,\sigma_0)$ is the self-shielded cross section in group i with boundaries $[E_i, E_{i+1}]$ for reaction x and nuclide j at background cross section σ_0 and temperature T. The flux $\Phi(E,T,\sigma_0)$ determines the type of Bondarenko factors that are generated. In the simplest case of a narrow resonance approach the flux is defined as:

$$\Phi(E,T,\sigma_0) = \frac{\Phi_{ref}(E)}{\sigma^j_x(E,T) + \sigma_0} \tag{83}$$

In other cases a more problem-dependent flux might be used to generate Bondarenko cross section data for the intermediate resonance approach. The Bondarenko data are usually stored as factors:

$$f^j_{x,i}(E,T,\sigma_0) = \frac{\tilde{\sigma}^j_{x,i}(E,T,\sigma_0)}{\sigma^j_x(E,T)} \tag{84}$$

To use the Bondarenko data to obtain shielded cross sections for transport calculations, the value of σ_0 is computed for the system of interest, and the corresponding factor $f^j_{x,i}(E,T,\sigma_0)$ is interpolated from tabulated values on the library. The shielded cross section is obtained by multiplying the shielding factor by the reference (i.e., infinite diluted) cross section. The salient feature of the Bondarenko method is that the flux weighting spectrum is parameterized by a function that depends only on the background cross section and temperature, which allows (82) to be pre-calculated for a specified set of background cross sections and temperatures that span the range of self-shielding conditions.

Bondarenko factors may also be given for the within group scattering, which is a measure of elastic scattering staying in the group. The within group scattering is defined as

$$\sigma^j_{removal}(E,T,\sigma_0) = \frac{1}{\Phi(E,T,\sigma_0)} \int_{E_i}^{E_{i+1}} dE \sigma^j_{elastic}(E,T)\Phi(E,T,\sigma_0) \int_{E_i}^{E} dE' P(E \to E', l=0) \tag{85}$$

where the definitions are as given in (82). Outside the thermal range, elastic scattering does not scatter to exit energies larger than the incident energy; therefore, the inner integral can be extended to the upper energy bound. This makes the removal cross section in the infinite diluted limit the same as the diagonal elements of the scattering transfer matrix for the elastic cross section at Legendre order zero.

Requirement 67: Bondarenko factors

67.1 The Bondarenko factors $f_{x,i}(E,T,\sigma_0)$ shall be stored in the reaction x's `<reaction>`, near the cross section they are derived from. This may actually be part of a `<crossSection>` element.

67.2 The Bondarenko factors are `<interp3d>` tables, given as a function of E, T and σ_0. The E dependence is grouped.

C. Production data

Production cross sections are used to store the total probability for producing a radioactive daughter, irrespective of what reaction or reactions were involved in creating that daughter. They are often used for modeling the activation of a material following irradiation. In ENDF they are also often used when a reaction can produce an isomer, to give the portion of the total cross section going to that isomer.

The production cross section σ_p for a specific product is a derived quantity that can be computed by:

$$\sigma_p = \sum_r \sigma_r \cdot M_r$$

where for each reaction r, σ_r is the cross section and M_r is the multiplicity of product 'p'.

Production cross sections are redundant and may be deprecated eventually, but they should continue to be supported for backwards-compatibility. The current solution (in GND) is to store production cross sections inside a 'production reaction' element. This is a special type of reaction that only contains a cross section and a single outgoing particle (the product).

Gas production is one specific form of production data that is used to understand the total amount of hydrogen and helium produced. The hydrogen gas production cross section is a sum of the proton, deuteron and triton production cross sections. The helium gas production cross section is a sum of the ^{3}He and α-particle production cross sections. The total gas production cross section is a sum of the hydrogen and helium gas production cross sections. In ENDF, gas production cross sections are given in MT's 203-207.

Requirement 68: Production data

68.1 Production (whether plain production or gas production) cross sections must be clearly marked so as not to cause confusion with regular cross sections

68.2 Production cross sections are stored in the appropriate `<reaction>`, along side the regular,

non-production, cross section data
68.3 Production cross sections clearly denote what data they are derived from

D. Radiation heating and damage

Radiation damage is obviously important in many nuclear applications. As an energetic particle transits a material, it ionizes atoms and molecules and dislocates atoms, heating the material. The modeling of the interactions is a complicated interplay of nuclear, atomic and materials physics. The two most commonly used metrics for characterizing material damage and heating are DPA and KERMA respectively.

Displacements per Atom (DPA) is a phenomenological measure of radiation damage:

$$\mathrm{DPA} = 0.8E_{\mathrm{avail}}/(2E_d) \tag{86}$$

where E_{avail} is the total energy available in a reaction and E_d is an element dependent parameter, usually equal to 25 eV, but can range as high as 90 eV.

KERMA is an acronym that means "Kinetic Energy Released in MAterials" and is defined as the sum of the initial kinetic energies of all the charged particles liberated by uncharged ionizing radiation (i.e., indirectly ionizing radiation such as photons and neutrons) in a sample of matter, weighted by the cross section of the reactions involved. For material i and reaction j, it can be computed by subtracting off the total energy released into outgoing neutrons and gammas and weighting by the appropriate reaction cross section:

$$k_{ij}(E) = \left(E + Q_{ij} - \overline{E}_{ijn} - \overline{E}_{ij\gamma}\right)\sigma_{ij}(E) \tag{87}$$

In the ENDF-6 format, KERMA can be stored in MT 301-450. The HEATR module of NJOY (MacFarlane, 2012) is responsible for calculating KERMA values.

Requirement 69: DPA and KERMA

69.1 KERMA shall be stored in the appropriate reaction `<product>`
69.2 DPA shall be stored in the appropriate `<reaction>`
69.3 Both KERMA and DPA will clearly denote what data they are derived from
69.4 Both KERMA and DPA will allow for storing uncertainty/covariance data

Radiation damage is also an under-developed part of the ENDF format. While radiation damage data needs to be archived, how it will fit within the hierarchy is a subject of future investigation. The reason is that traditional measures of damage such as displacements per atom (DPA) are crude and new approaches are being developed that better incorporate the materials properties. These approaches are being examined as part of a recently formed IAEA Coordinated Research Project (Stoller, 2012). In the future, we expect to need to draft requirements and specifications at the very least for damage cross section as well as other parameters such as the "athermal recombination-corrected DPA".

ACKNOWLEDGMENTS

We wish to acknowledge valuable review and feedback from all the contributors. For a full list, see appendix D. The work was performed by staff from several laboratories. The US DOE Office of Nuclear Physics sponsored work under Contracts DE-AC02-06CH11357 (Argonne National Laboratory), DE-AC02-05CB11231 (Lawrence Berkeley National Laboratory), DE-AC02-98CH10886 with Brookhaven Science Associates, LLC (Brookhaven National Laboratory), and DE-AC05-76OR00022 with UNIRIB (Oak Ridge National Laboratory). Other work was performed under the auspices of the National Nuclear Security Administration of the U.S. Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396 and at Lawrence Livermore National Laboratory under Contract No. DE-AC52-07NA27344. The project was partly funded through the Nuclear Data Program Initiative of the American Recovery and Reinvestment Act (ARRA).

Appendix A: Graphical Notation

In order to illustrate points and various proposed layouts of the top level hierarchy, we used diagrams written in the Crow's Foot notation for Entity Relationships (Barker, 1990) as implemented in the yEd code (yEd, 2015). The Crow's Foot notation is one type of diagram used to show entity relationships in the Unified Modeling Language (UML) (UML.org, 2015). The two tables detail the connections between elements (Table III) and the elements themselves (Table IV).

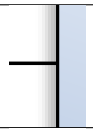
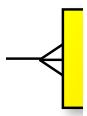
Symbol	Meaning
	Unspecified
	Exactly one
	Zero or one
	One or more
	Zero or more

TABLE III Connections denoting the number of child elements that are contained within a parent element within the Crow's Foot notation.

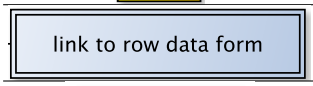
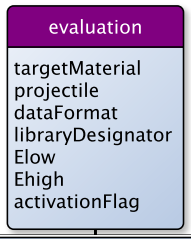
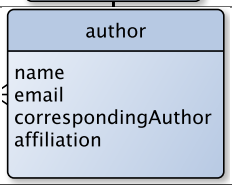
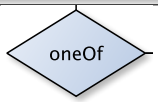
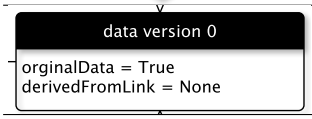
Symbol	Meaning
	Root node element
	High level element
	Mid level element
	Low level element
	Link to an element
	Root node element with attributes
	Element with attributes
	Condition
	Sample container

TABLE IV Crow's Foot notation elements. Any of these may be a parent element or contained within another element.

Name	Definition
E	Kinetic energy of the projectile.
$\sigma(E)$	Cross section as a function of E.
E'	Kinetic energy of an outgoing particle.
θ	Angular direction of outgoing particle relative to projectile's direction (i.e., $\hat{\mathbf{v}}' \cdot \hat{\mathbf{v}} = \cos(\theta)$).
$\mu = cos(\theta)$	Cosine of θ.
$f(\mu)$	Angular function for an outgoing particle. May not be normalized.
$f(E')$	Energy function for an outgoing particle. May not be normalized.
$f(E'\|E,\mu)$	Energy function for an outgoing particle for a given E and μ. Function may not be normalized.
$f(\mu\|E,E')$	Angular function for an outgoing particle for a given E and E'. Function may not be normalized.
$P(\mu\|E)$	Angular probability density function for an outgoing particle for a given E. Integration over μ is 1.
$P(E'\|E)$	Energy probability density function for an outgoing particle for a given E. Integration over E' is 1.
$P(\mu,E'\|E)$	Angular-energy probability density function for an outgoing particle for a given E. The data for a given E are ordered by μ and each μ has a function $f(E'\|E,\mu)$. Examples of this type of data in the ENDF-6 format include MF=6, LAW=1 and LANG=1,11-15. Integration over μ and E' yields 1.
$P(E',\mu\|E)$	Energy-angular probability density function for an outgoing particle for a given E. The data for a given E are ordered by E' and each E' has a function $f(\mu\|E,E')$. An example of this type of data in the ENDF-6 format is MF=6, LAW=7. This represents the same function as $P(\mu,E'\|E)$ but indicates that the data are stored in a different order. Integration over μ and E' yields 1.
$\frac{d\sigma(E,\mu)}{d\Omega}$	Differential cross section.
$\frac{d^2\sigma(E,\mu,E')}{d\Omega dE'}$	Double differential cross section.

TABLE V List of variables used throughout this document and their definition.

Appendix B: Definitions of integrated and differential cross sections

Table V lists some of the definitions of integrated and differential cross sections used in this document. Further terms are defined in the following appendix. The integrated and differential cross sections are related through:

$$\frac{d^2\sigma(E,\mu,E')}{d\Omega dE'} = \frac{\sigma(E)P(\mu,E'|E)}{2\pi} = \frac{\sigma(E)P(E',\mu|E)}{2\pi} \tag{B1}$$

$$\frac{d\sigma(E,\mu)}{d\Omega} = \frac{\sigma(E)P(\mu|E)}{2\pi} \tag{B2}$$

For two-body reactions, the double differential cross section in the center-of-mass can be written as

$$\frac{d^2\sigma(E,\mu,E')}{d\Omega dE'} = \frac{\sigma(E)\,P(\mu,E'|E)}{2\pi} = \frac{\sigma(E)\,P(\mu|E)\,\delta(E'-E'(E,\mu))}{2\pi} \tag{B3}$$

where $E'(E,\mu)$ can be calculated from kinematics and δ is the Dirac delta function. Hence, for two-body reactions only the angular probability density function in the center-of-mass is needed.

Appendix C: Terminology

A: The total number of protons and neutrons in a given nucleus. It is also known as the nucleon or mass number.

abundance: For isotopes that occur naturally, the abundance values are proportional to the probability of finding these isotopes and normalized so that the sum of the abundances for all the isotopes of a given chemical element is equal to 100 (Holden, 2004).

α **decay:** The emission of a ^{4}He nucleus (α particle).

α **particle:** A ^{4}He nucleus, that is, a nucleus made up of 2 neutrons and 2 protons.

AMPX: AMPX (Dunn, 2002) is a modular system of computer programs developed at ORNL with primary emphasis on processing neutron and photon evaluations to produce cross-section libraries for nuclear systems analysis.

application programming interface (API): The set of routines, protocols, and tools for building software applications. The API specifies how software components should interact.

atom: An atom is the smallest unit of matter that defines the chemical elements. Every solid, liquid, gas, and plasma is made up of neutral or ionized atoms.

atomic mass unit (amu): The atomic mass unit (amu) is defined so that 1 amu is equal to the mass of a ^{12}C atom divided by 12.

attribute, XML: Attributes are additional descriptive text added to XML elements, or to provide additional information about an element.

Auger electrons: Electrons that are produced when a vacancy in an orbit A is filled by an electron from the orbit B and an electron from an orbit C is ejected. These electrons are labeled by the three orbits that are involved in the production. For instance, AugerKL2L3 means that the original vacancy was in the K orbit, which was filled by an electron in the L2 orbit and the ejected electron came from the L3 orbit. Coster-Kronig transitions are a special type of Auger electrons where the last two orbit are part of the same shell.

Following nuclear decay, vacancies in the electron orbits are produced, which are filled by the emission of X-rays and electrons. Often, instead of listing the energy and intensity for each Auger electron, average intensities and sum intensities are given. For instance, the intensity of the Auger K electrons is the sum of the intensities for all the KBC Auger electrons.

β^- **decay:** The transformation of one neutron inside a nucleus into a proton plus an electron and an antineutrino: $n \rightarrow p + e^- + \bar{\nu}_e$

β^+ **decay:** The transformation of one proton inside a nucleus into a neutron plus a positron and a neutrino: $p \rightarrow n + e^+ + \nu_e$

β–**delayed particle emission:** The emission of a nucleon, nucleons or a nucleus following β-decay. For proton rich nuclei, the emission of a proton following β^+ decay and electron capture has been observed. For neutron rich nuclei, the release of one or two neutrons following β^- decay is possible. The emission of α particles has been observed for some nuclei in all types of β decay. Also, for a few nuclei, fission can take place following β^+ decay and electron capture.

branching ratio: The probability of a certain event occurring when multiple events are possible.

bremsstrahlung: Literally "braking radiation". The process of electromagnetic radiation when a charged particle is accelerated or decelerated.

BROND: The Russian Evaluated Neutron Data Library developed at Center Jadernykh Dannykh (CJD) in Obninsk, Kaluga Region, Russia. Further information available at `http://www.ippe.obninsk.ru/podr/cjd/`.

CALENDF: The CALENDF Nuclear Data Processing System is used to convert the evaluation defining the cross-section in ENDF format (i.e., the pointwise cross-sections and/or the resonance parameters, both resolved and unresolved) into forms useful for applications. Those forms used to describe neutron cross-section fluctuations correspond to "cross-section probability tables", based on Gauss quadratures and effective cross-sections. CALENDF also provides capabilities for group collapsing, for merging of several nuclei and for temperature interpolation; these calculations are based on data probability table description. CALENDF is developed by the Commissariat a l'Energie Atomique, Centre de Saclay.

CENDL: Chinese Evaluated Data Library is an evaluated nuclear reaction data library developed at the Chinese Nuclear Data Center (CNDC), Beijing, China in support of Chinese nuclear applications.

channel: is context sensitive concept. In resonance region (and anywhere else where we are using the R-matrix formalism), a channel has a specific meaning

as all the quantum numbers needed to uniquely denote a quantum state. For a two-body reaction, that usually means $c = \{\alpha, s, \ell, J\}$. For N-body reactions, we use the channel more loosely since in this cases many processes can lead to the same reaction products. For this reason, we will try to avoid using the term "channel" when discussing N-body reactions

cluster decay: The emission of a nucleus heavier than an α particle, for instance ^{14}C. Branching ratios for this decay tend to be very small, due to the large Coulomb barrier encountered by the cluster and its very small pre-formation factor, that is, the probability of finding the cluster formed inside the nucleus.

CoH: A modern object oriented nuclear reaction code developed by T. Kawano. In includes both a coupled channel solver as well as a models for compound nuclear and pre-equilibrium physics.

coherence, quantum mechanical: Two quantum mechanical wave functions are coherent if they have a constant phase difference and the same frequency (and therefore energy).

Compton scattering: Compton scattering is the inelastic scattering of a photon by a quasi-free charged particle, usually an electron. It results in a decrease in energy (increase in wavelength) of the photon (which may be an X-ray or gamma ray photon), called the Compton effect. In the ENDF format, is is termed incoherent elastic photoatomic scattering and the data stored is an incoherent scattering function which modifies Klein-Nishina formula.

conversion electrons (CE): An electron released from the atomic shell by transferring the energy of a gamma quantum emitted from the same nucleus to this electron. The kinetic energy of the conversion electron is equal to the energy of the gamma quantum reduced by the binding energy of the electron.

correlation matrix: A matrix that describes the correlations between parameters in a covariance matrix. The correlation matrix is defined in section IV.A, equation (17).

Coster-Kronig transition: The Coster-Kronig transition is a special case of the Auger process in which the vacancy is filled by an electron from a higher subshell of the same shell. If, in addition, the electron emitted (the "Auger electron") also belongs to the same shell, one calls this a super Coster-Kronig transition.

Coulomb scattering: See Rutherford scattering.

covariance matrix: A covariance matrix (also known as dispersion matrix or variance-covariance matrix) is a matrix whose element in the i, j position is the covariance between the i^{th} and j^{th} elements of a random vector (that is, of a vector of random variables).

covariance matrix, relative: The covariance matrix, scaled by the original data. See equation (14) for a precise definition.

cross section: If a beam of particles (or photons) enters a thin layer of material (thickness dx), then the particle number N will be reduced by $dN = -\mu N dx$, where μ is the attenuation coefficient. To describe the attenuation coefficient in a way independent of the material density, one introduces the cross section $\sigma = \mu/n$, where n is the numerical density (number of atoms per volume) of the material. σ has the dimension of an area; it expresses the likelihood of interaction between particles. σ depends on the incident energy of the particles in the beam and the energy dependence is denoted $\sigma(E)$.

cross section, capture: The capture cross section is the reaction cross section for the process in which a projectile is captured by a target and one or more photons is emitted.

cross section, differential: The cross section for ejecting a reaction product with energy E' into a direction $\Omega = (\theta, \varphi)$ is $d\sigma(E)/dE'd\Omega$.

cross section, elastic: Scattering cross section where the nucleus is unchanged in either isotopic composition or internal energy after interacting with a neutron.

cross section, fission: Cross section for the process where a neutron is captured and the nucleus fissions.

cross section, gas production: The sum of the production cross sections for all reactions which produce any isotope of hydrogen and helium nuclei.

cross section, inelastic: Scattering cross section where the nucleus is left in an excited state.

cross section, infinite dilute: The resonance integral (see Equation C15) where the scalar flux, $\phi(E)$ is assumed to be $1/E$.

cross section, flux weighted: The (integral) cross section convoluted with a (flux) weighting function $\phi(E)$:

$$\overline{\sigma} = \int dE\, \phi(E)\sigma(E). \tag{C1}$$

cross section, integral: See "cross section, integrated."

cross section, integrated: The differential cross section integrated over outgoing energy E' and solid angle Ω:

$$\sigma(E) = \int dE' d\Omega \, \frac{d\sigma(E)}{dE' d\Omega}. \tag{C2}$$

The integrated cross section is not defined for reactions with incident charged particles. The integrated cross section is often just called the "cross section".

cross section, macroscopic: The product of the atom density and the cross section.

cross section, Maxwellian averaged: A Maxwellian averaged cross section is a flux weighted cross section where the flux is a Maxwellian distribution:

$$\text{MACS}(kT) = \frac{2}{\sqrt{\pi}} \frac{a^2}{(kT)^2} \int_0^\infty dE E e^{-aE/kT} \sigma(E) \tag{C3}$$

where $a = m_{\text{targ}}/(m_{\text{proj}} + m_{\text{targ}})$

cross section, non-elastic: The total cross section minus the elastic cross section.

cross section, production: Given a set of cross sections that produce a given particle $a \in \{\gamma, n, p, d, t, ...\}$, $\{\sigma_{i,a}\}$, the production cross section is the multiplicity weighted sum of cross sections:

$$\sigma_{Xa}(E) = \sum_i M_{i,a}(E) \sigma_{i,a}(E) \tag{C4}$$

cross section, sum rule: Any addition rule that relates different cross sections for the same target–projectile system.

cross section, total: The sum of all reaction cross sections and the elastic cross section.

Crow's Foot Notation: A notation used in Entity-Relationship modeling. Crow's foot diagrams represent entities as boxes, and relationships as lines between the boxes. Different shapes at the ends of these lines represent the cardinality of the relationship.

cumulative distribution function (CDF): In probability theory and statistics, the cumulative distribution function (CDF), or just distribution function, describes the probability that a real-valued random variable x with a given probability distribution will be found to have a value less than or equal to x. In the case of a continuous distribution, it gives the area under the probability density function from minus infinity to x.

delayed fission gamma spectrum (DFGS): The energy spectrum of emitted gammas from the beta (and other) decays of the prompt fission fragments.

delayed fission neutron spectrum (DFNS): The energy spectrum of emitted neutrons from the beta (and other) decays of the prompt fission fragments.

Delbrück scattering : Delbrück scattering is the coherent elastic scattering of photons in the Coulomb field of heavy nuclei.

deuterium: A neutral atom with 1 electron and a deuteron as a nucleus. An isotope of the hydrogen atom.

deuteron: A ^{2}H nucleus, that is, a nucleus made up of 1 neutron and 1 proton. Not to be confused with deuterium.

Digital Object Identifier (DOI): A digital object identifier (DOI) is a character string (a "digital identifier") used to uniquely identify an object such as an electronic document. Metadata about the object is stored in association with the DOI name and this metadata may include a location, such as a URL, where the object can be found. The DOI for a document remains fixed over the lifetime of the document, whereas its location and other metadata may change. Referring to an online document by its DOI provides more stable linking than simply referring to it by its URL, because if its URL changes, the publisher need only update the metadata for the DOI to link to the new URL.

displacements per atom (DPA): Displacements per Atom (DPA) is a phenomenological measure of radiation damage. DPA $= 0.8E_{\text{avail}}/(2E_d)$ where E_{avail} is the total energy available in a reaction and E_d is an element dependent parameter, usually equal to 25 eV, but can range as high as 90 eV.

distribution, angular: Probability density function that gives the probability of a particle scattering in direction $\mu = cos(\theta)$ as a function of incident energy $E, P(\mu|E)$

distribution, energy: Probability density function that gives the probability of a particle scattering with an outgoing energy E' as a function of incident energy $E, P(E'|E)$

distribution, energy-angle distributions: Probability density function that gives the probability of a particle scattering in direction $\mu = cos(\theta)$ with an outgoing energy E' as a function of incident energy E, $P(E', \mu|E)$

dose: The product of a radiation energy times the probability per disintegration, the resulting unit is MeV×Bq-s.

double β decay: The simultaneous transformation of two neutrons inside a nucleus into two protons, or alternatively, the simultaneous transformation of two protons into two neutrons.

EADL: Evaluated Atomic Data Library, a library developed at LLNL to store atomic relaxation data. Atomic relaxation data describes the emission of electrons and photons as an atom relaxes back to neutrality following an ionization event. EADL is stored in the ENDL format and the contents of EADL have been translated into the ENDF format and are equivalent to the ENDF atomic relaxation sub-library.

EAF: The European Activation File (EAF) is the collection of nuclear data that is required to carry out inventory calculations of materials that have been activated following exposure to neutrons or charged particles.

EDA: R-matrix fitting code developed at LANL for fitting reactions on light nuclei.

EEDL: Evaluated Electron Data Library, a library that describes the interaction of electrons with matter. EEDL is stored in the ENDL format and the contents of EEDL have been translated into the ENDF format and are equivalent to the ENDF electron sub-library.

EFF: The European Fusion File (EFF) Project is a collaborative project with work funded by the European Fusion Development Agreement (EFDA). The emphasis is on the pooling of resources and removal of duplication of effort, leading to the efficient development of two types of nuclear data libraries for use in fusion power plant design and operation studies. The two branches consist of, on the one hand, a transport file for modeling and design capabilities and, secondly, an activation file for the calculation and simulation of dose rates and energy release during operation of a future power plant.

elastic scattering: A scattering event in which the scattering particle is re-emitted with an outgoing energy such that the scattering particle's incident and outgoing energies are equal in the system center of mass frame.

elastic scattering, compound: Elastic scattering in which the scattering particle is absorbed into the target nucleus, forming a compound nucleus which later decays, (re)emitting the scattering particle. Compound elastic scattering angular distributions are typically nearly isotropic in the rest frame of the colliding system.

elastic scattering, shape: Elastic scattering in which the scattering particle scatters off the target nucleus without being absorbed. Shape elastic scattering angular distributions are typically anisotropic in the rest frame of the colliding system.

elastic scattering, TSL: Elastic scattering in which a low energy/thermal neutron scatters either coherently or incoherently off many nuclei in a material.

electron: A negatively charged fundamental particle (lepton). It has a mass of 0.5109989(4) MeV, a charge of $-1.60217646(6) \times 10^{-19}$ Coulombs and $J^{\Pi} = 1/2^{+}$. Electrons bind to atomic nuclei, occupying specific atomic levels.

electron capture (EC): The process where one of the protons inside a nucleus following the interaction with one of the orbiting electrons transforms into a neutron plus a neutrino: $p + e^{-} \rightarrow n + \nu_e$

element, chemical: Materials with identical chemical properties. There is a one-to-one correspondence between the element name and the number of protons. For instance, all carbon atoms have 6 protons in their nuclei.

element, XML: Everything from (including) the element's start tag to (including) the element's end tag. An element can contain other elements, text, attributes, or a mix of these.

EMPIRE: Modular system of nuclear reaction codes for advanced modeling of nuclear reactions using various theoretical models. It consists of a number of FORTRAN codes, input parameter libraries, and experimental data library (EXFOR/CSISRS) - operated through the Graphic User Interface (GUI). Nuclear data evaluation is facilitated by the ENDF-6 formatting, file verification and graphical comparison with experimental data. It is available at `http://www.nndc.bnl.gov/empire/`.

ENDF/A: Formerly the development library of the ENDF project. Historically partial and in development evaluations were maintained in the ENDF/A library.

ENDF/B: The main release library of the ENDF project. The ENDF/B library is a product of the Cross Section Evaluation Working Group (CSEWG), a long standing collaboration of institutions within and without the United States.

ENDF format: A nuclear data format used by many nuclear data projects including the namesake library, the ENDF/B library. The ENDF format is

maintained by the Cross Section Evaluation Working Group (CSEWG).

ENDL: LLNL's Evaluated Nuclear Data Library, an evaluated nuclear reaction data library comparable in scope with the ENDF library. ENDL is used to support transport calculations at LLNL.

ENDL format: The format of data in ENDL. It is much simpler than the ENDF file, but is much less expressive than the ENDF format as all data must be stored as a multi-dimensional linear interpolation table.

end point energy: The maximum kinetic energy that an electron in β^- decay or a positron in β^+ decay can have, obtained when the kinetic energy of the neutrino/anti-neutrino is equal to zero.

energy group: A range between two energy points over which the transport equation is integrated.

energy loss: See stopping power.

entity–relationship model: An entity–relationship model (ER model) is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as a relational database.

ENSDF: The Evaluated Nuclear Structure Data File. ENSDF contains evaluated nuclear structure and decay data in a standard format. For each nuclide, all known experimental data used to deduce nuclear structure information are included. Each type of experiment is presented as a separate dataset. In addition, there is a dataset of "adopted" level and gamma-ray transition properties, which represent the evaluator's determination of the best values for these properties, based on all available experimental data.

ENSDF format: The format used to store the ENSDF database.

EPDL: Evaluated Photon Data Library, a library developed at LLNL to store photo-atomic data. EPDL is stored in the ENDL format and the contents of EPDL have been translated into the ENDF format and are equivalent to the ENDF photo-atomic sublibrary.

equiprobable bins: Discretized bins spaced such that each bin has the same probability.

Ericson fluctuations: Ericson fluctuations are the statistical fluctuations observed in cross sections in the overlapping resonance region (ORR).

evaluate: The process of digesting experimental data, combining it with the predictions of nuclear model calculations and attempting to extract the true value of a cross section is referred to as an evaluation.

evaluation: One projectile and one target material and all the data needed to describe the interactions between the two over some incident energy range. The projectile is usually assumed to impinge upon a stationary target and is usually assumed to be less massive than the target material. However, the data structure must be flexible enough to store data in "inverse kinematics" where the lighter particle is at rest relative to the heavier one or in the center of mass frame. The data structure must also be flexible enough that the target material is actually a nontrivial collection of nuclei such as in thermal neutron scattering.

evaluator: One who evaluates.

EXDL: The Evaluated Excitation Data Library, a library that describes the excitation of atoms due to photon interaction. ENDF does not yet have a format to support this data.

eXtensible Markup Language (XML): A markup language that defines a set of rules for encoding documents in a format which is both human-readable and machine-readable.

flux weighting: The convolution of a quantity, such as cross section $\sigma(E)$, with a (flux) weighting function $\phi(E)$:

$$\overline{\sigma} = \int dE \phi(E) \sigma(E). \tag{C5}$$

fission: Nuclear fission is either a nuclear reaction or a radioactive decay process in which the nucleus of an atom splits into smaller parts (lighter nuclei). The fission process often produces free neutrons and photons (in the form of gamma rays), and releases a very large amount of energy even by the energetic standards of radioactive decay.

fission, spontaneous: A radioactive decay process which results in fission.

fission product yield (FPY): The yield of a particular isotope some time after a fission event. As there are two fission fragments following every (binary) fission event, the yield sums to two.

fission product yield, cumulative (CFPY): The yield of isotopes after the fission fragments have undergone all (beta) decays and are now stable.

fission product yield, independent (IFPY): The yield of a particular fission fragment immediately after a fission event. As there are two fission fragments following every (binary) fission event, the yield sums to two.

Fudge: An open source processing and data modification code developed at LLNL. Fudge is the first processing code capable of manipulating data in both the ENDF and GND format. Fudge originally stood for "For Updating Data and Generating Evaluations".

gamma ray: The term "gamma rays" is used here for electromagnetic waves (photons) that have a nuclear origin, that is, photons are emitted following the rearrangement of the protons and neutrons in a nucleus. In contrast, the term "X rays" is used for photons emitted following the rearrangement of the electrons orbiting an atomic nucleus. Gamma rays are one type of radiation.

gamma ray emission: Atomic nuclei are quantum system with a discrete set of energies. Gamma ray emission can take place when a nucleus rearranges its protons and neutrons into a lower energy state.

GNASH: GNASH is a comprehensive nuclear reaction modeling code developed at LANL, comparable to EMPIRE in scope and functionality.

GND: Generalized Nuclear Data format. The name for the hierarchical arrangement of nuclear data developed at LLNL. It was originally developed in XML, but can be serialized into any hierarchical low level data format such as HDF5. GND is a prototype of the data structure being developed by WPEC SubGroup 38.

group: See "energy group".

half-life ($T_{1/2}$): The length of time for a given radioactive species to reduce its activity in half. The number of decays as a function of time is given as:

$$N(T) = N_0 \exp\left(-\ln(2) \times (T - T_0)/T_{1/2}\right) \qquad \text{(C6)}$$

$T_{1/2}$ is related to the life-time τ by:

$$T_{1/2} = \ln(2) \times \tau \qquad \text{(C7)}$$

and to the width Γ by

$$T_{1/2} = \ln(2) \times (h/2\pi)/\Gamma \qquad \text{(C8)}$$

where h is Planck's constant. Please note that when the half-life of a given nuclear level is smaller than 10^{-15} seconds, it is customary to list the width (Γ) of the level instead.

HDF5: Hierarchical Data Format, version 5 (HDF5) is a data model, library, and file format for storing and managing data. It supports an unlimited variety of datatypes, and is designed for flexible and efficient I/O and for high volume and complex data.

hindrance factor (HF): The ratio between the alpha decay partial experimental half-life and a calculated half-life.

incoherence, quantum mechanical: Two quantum mechanical wave functions are incoherent if they either do not have a constant phase difference or the same frequency (and therefore energy). Quantum mechanical wavefunctions (or parts of the same wavefunction) are said to be incoherent if their phases are not equal.

intensity, radiation: Radiation intensities indicate the probability of observing the corresponding radiations. Two different conventions are used:

- For decay radiation, intensities are listed per 100 decays of the parent nucleus. For instance, in the decay of ^{232}Th, the alpha particle with 4012 keV is listed as having an intensity of 78.2%, which means that this alpha particle will be emitted 78.2 times for every 100 decays of 232Th.
- For the gamma rays, intensities correspond to gamma branching ratios for each level, assigning 100 to the strongest gamma ray.

IBANDL: The Ion Beam Analysis Nuclear Data Library developed and formerly maintained by A. Gurbich under the IAEA auspices. It contains available experimental nuclear cross-sections relevant to Ion Beam Analysis and is available at `https://www-nds.iaea.org/exfor/ibandl.htm`.

inelastic scattering: A scattering event in which the scattering particle is reemitted with an outgoing energy different from the scattering particle's incident energies in the system center of mass frame. Usually the scattering particle's energy is lost in the transition, but if the target is already in an excited state it is possible for the scattering particle to gain energy from the de-exciting target nucleus. This is occasionally referred to as "super–elastic scattering."

inelastic TSL scattering: Inelastic scattering in which a low energy/thermal neutron scatters either coherently or incoherently off many nuclei in a material.

internal conversion: A process where the transition from one nuclear level to another level in the same nucleus is carried out by transferring the excess energy to an orbiting electron. The alternate process

is gamma emission. The electrons are ejected from the atom with an energy equal to the transition energy minus the electron binding energy. These electrons are called conversion electrons and are labeled by the orbit the electron had occupied. For instance, CE-L means conversion electron from the L shell.

internal conversion coefficient (ICC): The ratio of the number of internal conversion decays to the number of gamma decays is the internal conversion coefficient, denoted α. The ICC can be decomposed into the ICC for individual transitions (electric vs. magnetic and by multipolarity). The probability of internal conversion P_{IC} is related to the probability of gamma emission P_γ by $P_{IC} = \alpha P_\gamma$.

inverse kinematics: A two-body collision in the lab frame in which the projectile is the more massive of the two bodies and the target is the less massive of the two bodies.

isobar: A number of nuclei with the same number of protons plus neutrons are called isobars, such as ^{144}Sm, ^{144}Nd and ^{144}Gd.

isomer: A nuclear isomer is a metastable excited state of an atomic nucleus.

isomeric Transition (IT): The process where a long-lived excited nuclear level decays by gamma emission or internal conversion.

isotone: A number of nuclei with the same number of neutrons are called isotones, such as ^{144}Sm, ^{142}Nd and ^{146}Gd.

isotope: A number of nuclei with the same number of protons are called isotopes, such as ^{144}Sm, ^{142}Sm and ^{146}Sm.

JEF: Joint Evaluated File, predecessor of the JEFF project.

JEFF: The Joint Evaluated Fission and Fusion File (JEFF) project is a collaboration between NEA Data Bank member countries. The JEFF library combines the efforts of the JEFF and EFF/EAF Working Groups to produce a common sets of evaluated nuclear data, mainly for fission and fusion applications. Available at `https://www.oecd-nea.org/dbdata/jeff/`.

JENDL: The Japanese Evaluated Nuclear Data Library, developed by the Japan Atomic Energy Agency (JAEA) to support nuclear application development in Japan. Available at `http://wwwndc.jaea.go.jp/jendl/j40/j40.html`.

J^Π: The angular momentum (J) and parity (Π) associated with a nuclear level or a particle. For instance, the ground states of nuclei with even number of protons and neutrons have $J^\Pi = 0^+$. The intrinsic J^Π of the proton is equal to $1/2^+$.

KERMA: An acronym that means "Kinetic Energy Released in MAterials" and is defined as the sum of the initial kinetic energies of all the charged particles liberated by uncharged ionizing radiation (i.e., indirectly ionizing radiation such as photons and neutrons) in a sample of matter, weighted by the cross section of the reactions involved. For material i and reaction j, it can be computed by subtracting off the total energy released into outgoing neutrons and gammas and weighting by the appropriate reaction cross section:

$$k_{ij}(E) = \left(E + Q_{ij} - \overline{E}_{ijn} - \overline{E}_{ij\gamma}\right)\sigma_{ij}(E) \qquad \text{(C9)}$$

Klein-Nishina formula: The Klein–Nishina formula gives the differential cross section of photons scattered from a single free electron in lowest order of quantum electrodynamics. At low frequencies (e.g., visible light) this is referred to as Thomson scattering; at higher frequencies (e.g., x-rays and gamma-rays) this is referred to as Compton scattering.

large angle Coulomb scattering (LACS): Charged particle scattering at large angles (typically $\gtrsim 10°$) described purely with Coulomb scattering. At low angles or at low energies, charge particle scattering must be treated with an alternate approach.

lethargy: Neutron lethargy, or logarithmic energy decrement, u, is a dimensionless logarithm of the ratio of the energy of source neutrons to the energy of neutrons after a collision:

$$u = \ln(E_o/E), \text{ or, } u_2 - u_1 = \ln(E_1/E_2). \qquad \text{(C10)}$$

Also, the feeling one has after working on a requirements document for far too long.

level: A quantum mechanical system or particle that is bound can only take on certain discrete values of energy.

logft: For each decay branch in β decay, the decay probability, $T_{1/2}$ and energy released can be combined in a single quantity known as logft, which is defined as:

$$\text{logft} = \log(f(Z, E_0)T_{1/2}) \qquad \text{(C11)}$$

where E_0 is the end-point energy for the transition and $f(Z, E_0)$ is the Fermi integral. Logft values increase with increasing $T_{1/2}$, decay probability and E_0 values. There is a correlation between the type of transition and its logft value.

MAT: The designator used in the ENDF format to distinguish materials.

material: In nuclear data, a material generalizes the concept of a target and may include a single nucleus or nucleon or a macroscopic collection of nuclei that a projectile scatters off of.

mass: Mass is a property of a material which determines resistance to being accelerated by a force and the strength of its mutual gravitational attraction with other bodies.

mass, nuclear: The mass of a nucleus with Z proton and N neutrons in a neutral-atom state is:

$$\begin{aligned}&\text{Mass}(Z,N) = \\ &\quad Z * \text{Mass(hydrogen atom)} \\ &\quad + N * \text{Mass(free neutron)} - BE(Z,N)\end{aligned} \tag{C12}$$

where $BE(Z,N)$ is the Binding Energy, that is, the energy needed to dissociate the nucleus into free nucleons. Note in this product, masses are given in energy units.

mass excess (Δ): The mass excess $\Delta(Z,N)$ is defined as:

$$\Delta(Z,N) = (\text{Mass}(Z,N)(\text{amu}) - A) \times \text{amu} \tag{C13}$$

where $\text{Mass}(Z,N)$ (amu) is mass in atomic mass units and $\text{amu} = \text{Mass}(6,6)/12$.

MF: In the ENDF format, the MF designator denotes the observable of interest. For example, MF=3 denotes cross section data.

Mott scattering: The modification of Rutherford scattering of an electron and a nucleus to include the effects of nuclear recoil and the nuclear magnetic moment.

MT: In the ENDF format, the MT designator denotes the reaction of interest. For example, MT=18 denotes fission.

multi-band treatment: An alternative scheme to improve group averaged data by dividing up the cross section within a group into bands. These bands are equivalent to cross section probability tables, if the cross section probabilities are given as a histogram. Typically the band are chosen to preserve the average cross section, the totally shielded, flux weighted cross section and the totally shielded current weighted cross section.

multifragmentation: Nuclear (multi-)fragmentation, defined as the nuclear decay mechanism in which at least three intermediate mass fragments ($Z \geq 3$) are produced, is the disassembly phenomenon specific to hot nuclear matter produced in nuclear collisions at beam energies of 20-100 MeV/nucleon.

multiplicity: The average number of particles emitted from a reaction per unit incident energy. For an $(n, 2n)$, the multiplicity of neutrons is 2 for all energies, but the photon multiplicity is variable and depends on the nuclear structure of the residual nucleus.

multipolarity: A measure of the angular momentum carried away by a photon during an electromagnetic transition or decay. An $\ell = 1$ transition is a dipole transition, $\ell = 2$ transition is a quadrupole transition, etc.

NJOY: LANL's nuclear data processing code, see `http://t2.lanl.gov/nis/codes/NJOY12/`.

nubar, $\bar{\nu}$: The average number of neutrons emitted during a fission reaction.

nubar, delayed, $\bar{\nu}_d$: The average number of neutrons emitted from the (usually beta) decay of the fission fragments. In the ENDF format, the delayed nubar is broken out into six time groups.

nubar, prompt, $\bar{\nu}_p$: The average number of neutrons emitted immediately after fission, due either to emission before or during the fission event or evaporated from the fission products immediately after fission.

nucleus: The very dense region consisting of protons and neutrons at the center of an atom.

pair production: Pair production is the creation of an elementary particle and its antiparticle, for example an electron and its antiparticle, the positron.

particle: A particle is any small, localized object that can be attributed properties such as mass, charge, spin, parity, and halflife. This definition is deliberately broad to include fundamental particles such as electrons or photons, composite particles such as atomic nuclei or atoms, and also excited states of composite particles. We recognize that this definition may be surprising to nuclear physicists who tend to think of nuclei as a separate family from fundamental particles.

photoexcitation: Photoexcitation is the physical process in which an electron in an atom or molecule is promoted to an excited state from the interaction of a photon.

photoionization: Photoionization is the physical process in which an ion is formed from the interaction of a photon with an atom or molecule.

pointwise: A function represented by an interpolation table (usually with lin-lin interpolation) is a pointwise function.

primary knock-on atom (PKA): For high values of the energy transferred, the target atom is called the primary knocked-on atom (PKA). The PKA interacts with the other atoms along its track. For each interaction, it transfers energy on the secondary target. Each subsequent target will similarly interact with other atoms and a cascade of interactions occurs. The result is a large number of atoms displaced (up to a few hundreds), in a "displacement cascade."

principal component analysis (PCA): A statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. For a multivariate Gaussian probability distribution, the orthogonal transform is constructed from the matrix of eigenvectors of the covariance matrix and the eigenvalues of the covariance matrix are the uncorrelated variables known as the principal components.

probability density function (PDF): In probability theory, a probability density function (PDF), or density of a continuous random variable, is a function that describes the relative likelihood for this random variable to take on a given value.

probability density function, Gaussian: The Gaussian PDF is given by:

$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{C14}$$

The parameter μ in this definition is the mean or expectation of the distribution (and also its median and mode). The parameter σ is its standard deviation; its variance is therefore σ^2. A random variable with a Gaussian distribution is said to be normally distributed and is called a normal deviate.

probability density function, log-normal: A log-normal (or lognormal) distribution is a continuous probability distribution of a random variable whose logarithm is normally distributed. Thus, if the random variable x is log-normally distributed, then $y = \ln(x)$ has a normal distribution. Likewise, if y has a normal distribution, then $x = \exp(y)$ has a log-normal distribution. A random variable which is log-normally distributed takes only positive real values.

probability density function, normal: See probability density function, Gaussian.

projectile: In a two-body collision in the laboratory frame, the projectile is the body in motion. The projectile is typically the less massive of the two colliding particles.

prompt fission gamma spectrum (PFGS): The spectrum of gammas emitted during and immediately after a fission event, but before the fission fragments undergo weak decays.

prompt fission neutron spectrum (PFNS): The spectrum of neutrons emitted during and immediately after a fission event, but before the fission fragments undergo weak decays.

PURM: The module in the Oak Ridge National Laboratory processing code AMPX (Dunn, 2002) responsible for producing cross section probability tables in the URR.

PURR: The module in the Los Alamos National Laboratory processing code NJOY (MacFarlane, 2012) responsible for producing cross section probability tables in the URR.

Q matrix: The matrix that connects the Cumulative Fission Yields with the Independent Fission Yields.

quantum number: Quantum numbers describe values of conserved quantities in the dynamics of a quantum system. Typical quantum numbers encountered in nuclear physics are spin, parity, orbital and total angular momenta, mass, charge and isospin.

Q value: The Q value for a reaction is the amount of energy released by that reaction.

Raleigh scattering: Rayleigh scattering named after the British physicist Lord Rayleigh, is the (dominantly) elastic scattering of light or other electromagnetic radiation by particles much smaller than the wavelength of the radiation. When the scattering is coherent one uses Thompson scattering formula modified by (anomalous) form factors

reaction: A nuclear reaction is a process in which two (or more) nuclei or nuclear particles collide, producing a different set of products than the initial particles. From this perspective, elastic scattering is not a "reaction" while inelastic scattering is a "reaction" as the energy of the inelastically scattered particle has changed.

reaction, exclusive: A reaction with well defined non-gamma reaction product multiplicities (e.g., (n,2n)). Inelastic reactions to discrete states are considered exclusive since the residual nucleus is left in a well defined state before it gamma cascades.

reaction, inclusive: A reaction that is a sum of exclusive reactions (e.g., total or fission). Inelastic from the continuum is not considered here since the residual is still well defined but total inelastic is inclusive since it is a sum of inelastic continuum and discrete reactions.

reaction, photonuclear: A nuclear reaction induced by a high energy photon (typically with incident energy $\gtrsim 1$ MeV)

Reference Input Parameter Library (RIPL): A common library of input parameters for use in nuclear reaction modeling codes. Much of the data in RIPL is derived from other sources (e.g., ENSDF) or from systematics. More information is available at `https://www-nds.iaea.org/RIPL-3/`.

resonance: In the context of nuclear data, a resonance is a compound state formed when a nucleus absorbs a projectile. The compound state is unstable against particle emission and so is characterized by emission widths.

resonance integral: The total effective absorption rate per atom is defined as the integral over the energy range of a resonance (ΔE_{res}):

$$I = \int_{\Delta E_{res}} \sigma_a(E)\phi(E)dE \tag{C15}$$

resonance region (RR): The region in a cross section where resonances are observed is called the resonance region (RR).

resonance region, resolved (RRR): The resolved resonance region is the portion of the resonance region where the resonance widths are small enough and the resonances are spaced far enough apart in energy that they can be individually resolved experimentally.

resonance region, overlapping (ORR): The overlapping resonance region is the portion of the resonance region at the highest energies where the resonance widths are larger or comparable to the inter-resonance spacing. Thus, the cross sections exhibit large interference effects and sizable fluctuations. This fluctuations are known as Ericson fluctuations.

resonance region, unresolved (URR): In the unresolved resonance region, the resonances of a cross section are so close together that, although they are not overlapping, they cannot be experimentally resolved.

R-matrix: A tool in computational quantum mechanics for studying two-body scattering. R-matrix theory begins by placing a reaction zone inside a spherical box. Outside the box, asymptotic (and thus calculable) wave functions are used to describe the pair of scattering particles. Using the continuity of flux on the boundary, an R-matrix can be defined which can be used to parameterize things such as the scattering cross section. The R-matrix method was originally formulated for studying resonances in nuclear scattering by Wigner and Eisenbud.

Rutherford scattering: The scattering of two charged particles purely by the static electric force. The Rutherford cross section is calculable either quantum mechanically and classically.

$S_{\alpha\beta}$: $S_{\alpha\beta}$ is the scattering kernel of the double differential elastic scattering cross section for thermal neutrons in the Thermal Scattering Law formalism.

SAMMY: A resonance fitting code based on R-matrix theory developed by ORNL. It is capable of fitting resonances on nearly all nuclei and is capable of simultaneously fitting hundreds if not thousands of resonances simultaneously.

self-shielding: Self-shielding occurs when the neutron flux in one part of a material is shielded from another part of the same material.

shell: A nuclear or atomic shell is a state or a collection of states with the same well defined quantum numbers. Electronic shells are usually denoted with the X-ray notation (e.g., K, L, M, N, ...) while nuclear shells are denoted using a different notation (e.g., s, p, d, f, ...).

shell model: The nuclear shell model is a model of the atomic nucleus which uses the Pauli exclusion principle to describe the structure of the nucleus in terms of energy levels.

spallation: The process in which a nucleus emits a large number of nucleons as a result of being hit by a high-energy particle, thus greatly reducing its atomic weight.

spin group: The name of the group of resonances with the same total spin and parity. Organizing resonances by spin groups results in much smaller tables of resonance parameters as the needless storing of zero widths is avoided.

stopping power: The retarding force acting on charged particles due to interaction with matter, resulting in loss of particle energy. The stopping power of the material is numerically equal to the loss of energy E per unit path length, x: $S(E) = -dE/dx$

subshell: Each shell in a shell model is composed of one or more subshells, which are themselves composed of atomic or nuclear orbitals.

TALYS: TALYS is a nuclear reaction modeling code developed as a collaboration between NRG, Petten and CEA. TALYS is comparable in scope and functionality to EMPIRE.

target: In a two-body collision in the laboratory frame, the target is the body at rest. The target is typically the more massive of the two colliding particles.

TENDL: TALYS Evaluated Nuclear Data Library is an evaluated nuclear data library generated using the TALYS reaction code. TENDL is available at `http://www.talys.eu/tendl-2012/`.

thermal scattering law (TSL): Thermal scattering law data describe the situation in which the de Broglie wavelength of an incident neutron is so large that the neutron wave function cannot resolve individual nuclei, but rather sees the macroscopic material.

Thompson scattering: Thomson scattering is the elastic scattering of electromagnetic radiation by a free charged particle, as described by classical electromagnetism. It is just the low-energy limit of Compton scattering.

time group: Following fission, many possible pairs of fission products are created each with their own distinct beta decay pathways and halflives. As there are many potential decay pathways and halflives, the ENDF-6 format allows the evaluator to lump pathways with similar halflives together into a maximum of 6 time-groups. A delayed fission neutron spectra can be generated for each time group.

transfer matrix: The transfer matrix is a matrix that arises during the discretization of the Boltzmann transport equation. See subsection VI for a precise definition.

UKNDL: United Kingdom Nuclear Data Library, the nuclear data library and format developed by UK's Atomic Weapons Establishment. UKNDL format was the prototype of both the ENDF and ENDL formats.

Unified Modeling Language (UML): The Unified Modeling Language is a general-purpose modeling language in the field of software engineering, which is designed to provide a standard way to visualize the design of a system.

Universal Resource Locator (URL): A URL is one type of Uniform Resource Identifier (URI); the generic term for all types of names and addresses that refer to objects on the World Wide Web. The term "Web address" is a synonym for a URL that uses the HTTP / HTTPS protocol.

uncertainty: The uncertainty is the square root of expected deviation of a measurement from the expected mean.

vacancy, shell: A (sub)shell vacancy occurs when a particle is knocked out of a nuclear or atomic shell, leaving a "hole" or unoccupied state in the shell.

Westcott factor: The ratio of the room temperature ($T = 293°\text{K}$) Maxwellian averaged cross section with the cross section evaluated at $E = 0.0253$ eV. If a cross section varies like $1/v$, then this ratio will be one.

Working Party on Evaluation Cooperation (WPEC): A framework provided by the Nuclear Energy Agency so that other institutions can co-operate and conduct multi-year projects which promote the exchange of information on nuclear data evaluations, measurements, nuclear model calculations, validation, and related topics.

xlink: XML Linking Language, or XLink, is an XML markup language and W3C specification that provides methods for creating internal and external links within XML documents, and associating metadata with those links.

xpath: XPath, the XML Path Language, is a query language for selecting nodes from an XML document. In addition, XPath may be used to compute values (e.g., strings, numbers, or Boolean values) from the content of an XML document.

xsd: XSD (XML Schema Definition), a recommendation of the World Wide Web Consortium (W3C), specifies how to formally describe the elements in an Extensible Markup Language (XML) document. It can be used by programmers to verify each piece of item content in a document.

xsdir: The xsdir file, serves as a table of contents for the transport code MCNP with information on where and how the data are stored for each target material can be found.

xsl: Extensible Stylesheet Language (XSL) is used to refer to a family of languages used to transform and render XML documents.

ZA: A common nucleus designator usually computed as $ZA = Z \times 1000 + A$.

Appendix D: Contributors

Here we attempt to list everyone who contributed to this document and the general Subgroup 38 discussions. We apologize if we have omitted anyone. Below we list everyone in English alphabetical order by family name. Each contributors affiliation is given in the form "institute, country". Where an institute has multiple locations, the form of the affiliation is "institute, city, country".

Contributor	Affiliation
Pascal Archier	*CEA, Cadarache, DEN/DER/SPRC, France*
Bret Beck†	*Lawrence Livermore National Laboratory, USA*
David Brown†‡	*Brookhaven National Laboratory, USA*
Oscar Cabellos◇	*Nuclear Energy Agency, Paris, France*
Roberto Capote Noy	*International Atomic Energy Agency, Vienna, Austria*
Chiba Go	*Hokkaido University, Japan*
Cho Young-Sik	*Korean Atomic Energy Research Institute, S. Korea*
Jeremy Lloyd Conlin†	*Los Alamos National Laboratory, USA*
Mark Cornock	*Atomic Weapons Establishment, Aldermaston, UK*
Mireille Coste-Delclaux	*CEA, Saclay, DEN/DM2S/SERMA, France*
D.E. (Red) Cullen	*Lawrence Livermore National Laboratory, USA (retired)*
Michael Dunn	*Oak Ridge National Laboratory, USA*
Emmeric Dupont◇	*NEA and CEA, Saclay, DSM/IRFU/SPhN, France*
Ulrich Fischer	*Karlsruhe Institute of Technology, Germany*
Robin Forrest	*International Atomic Energy Agency, Vienna, Austria*
Fukahori Tokio	*Japanese Atomic Energy Agency, Tokai, Japan*
Ge Zhigang	*Chinese Nuclear Data Center, Beijing, China*
Wim Haeck	*Institute for Radiological Protection and Nuclear Safety, France*
Ayman Hawari	*North Carolina State University, USA*
Michal Herman	*Brookhaven National Laboratory, USA*
Jesse Holmes†	*Bettis Atomic Power Laboratory, USA*
Ishikawa Makoto	*Japanese Atomic Energy Agency, Tokai, Japan*
Iwamoto Osamu	*Japanese Atomic Energy Agency, Tokai, Japan*
Robert Jacqmin	*CEA, Cadarache, DEN/DER/SPRC, France*
Timothy Johnson	*Brookhaven National Laboratory, USA*
Cedric Jouanne	*CEA, Saclay, DEN/DM2S/SERMA, France*
A.C. (Skip) Kahler	*Los Alamos National Laboratory, USA*
Kim Do Heon	*Korean Atomic Energy Research Institute, S. Korea*
Ivo Kodeli	*Jozef Stefan Institute, Slovenia*
Arjan Koning†	*International Atomic Energy Agency, Vienna, Austria*
Konno Chikara	*Japanese Atomic Energy Agency, Tokai, Japan*
Kugo Teruhiko	*Japanese Atomic Energy Agency, Tokai, Japan*
Luiz Leal	*Oak Ridge National Laboratory, USA*
Morgan Lee	*Culham Centre for Fusion Energy, UK*
Cecil Lubitz	*Knolls Atomic Power Laboratory, USA (retired)*
Fausto Malvagi	*CEA, Saclay, DEN/DM2S/SERMA, France*
Caleb Mattoon†	*Lawrence Livermore National Laboratory, USA*

* WPEC Subgroup 38 Co-ordinator
† Requirements document author
‡ Requirements document editor
◇ NEA contact

Contributor	Affiliation
Dennis McNabb*†	*Lawrence Livermore National Laboratory, USA*
Robert Mills†	*National Nuclear Laboratory, UK*
Nishihara Kenji	*Japanese Atomic Energy Agency, Tokai, Japan*
Gilles Noguere	*CEA, Cadarache, DEN/DER/SPRC, France*
Marco Pigni	*Oak Ridge National Laboratory, USA*
Yannick Peneliau	*CEA, Cadarache, DEN/DER/SPRC, France*
Paul Romano	*Knolls Atomic Power Laboratory, USA*
Danila Roubtsov†	*CNL, Chalk River Laboratories, Canada*
Cyrille de Saint-Jean	*CEA, Cadarache, DEN/DER/SPRC, France*
Sato Tatsuhiko	*Japanese Atomic Energy Agency, Tokai, Japan*
Anthony Scopatz	*University of Chicago, USA*
Valentin Sinitsa	*Kurchatov Institute, Russia*
Alejandro Sonzogni	*Brookhaven National Laboratory, USA*
Nicolas Soppera	*Nuclear Energy Agency, OECD, Paris, France*
Jean-Christophe Sublet	*Culham Centre for Fusion Energy, UK*
Suyama Kenya	*Japanese Atomic Energy Agency, Tokai, Japan*
Patrick Talou†	*Los Alamos National Laboratory, USA*
Jean-Christophe Trama	*CEA, Saclay, DEN/DM2S/SERMA, France*
Andrej Trkov	*International Atomic Energy Agency, Vienna, Austria*
Romona Vogt†	*Lawrence Livermore National Laboratory and University of California Davis, USA*
Dorothea Wiarda	*Oak Ridge National Laboratory, USA*
Morgan C. White†	*Los Alamos National Laboratory, USA*
Yokoyama Kenji	*Japanese Atomic Energy Agency, Tokai, Japan*
Viktor Zerkin†	*International Atomic Energy Agency, Vienna, Austria*

* WPEC Subgroup 38 Co-ordinator
† Requirements document author
‡ Requirements document editor
◇ NEA contact

REFERENCES

R. Barker, *CASE Method: Entity Relationship Modeling*. Reading, MA: Addison-Wesley Professional. ISBN 0-201-41696-4. http://www.entitymodelling.org/ (1990).

B. Beck, private communication (2005).

J. Beringer, *et al.*, "Monte Carlo Particle Numbering Scheme", Phys. Rev. D **86**, 010001 (2012).

J.M. Blatt, L.C. Biedenharn, Rev. Mod. Phys. **30**, 258 (1958).

D.A. Brown *et al.*, "Uncertainty quantification in the Nuclear Data Program", J. Phys. G: Nucl. Part. Phys. **42**, 034020 (2015).

C.R. Brune, "Alternative parameterization of R-matrix theory", Phys. Rev. C **66**, 044611 (2002)

Handbook of Nuclear Engineering, Volume 1: Nuclear Engineering Fundementals, ed. D.G. Cacuci, Spinger, ISBN 978-0-387-98130-7 (2010).

R. Capote, M. Herman, P. Oblozinsky, P.G. Young, S. Goriely, T. Belgya, A.V. Ignatyuk, A.J. Koning, S. Hilaire, V.A. Plujko, M. Avrigeanu, O. Bersillon, M.B. Chadwick, T. Fukahori, Zhigang Ge, Yinlu Han, S. Kailas, J. Kopecky, V.M. Maslov, G. Reffo, M. Sin, E.Sh. Soukhovitskii, P. Talou, Nucl. Data Sheets **110**, pp. 3107-3214 (2009).

A.D. Carlson, V.G. Pronyaev, D.L. Smith, N.M. Larson, Zhenpeng Chen, G.M. Hale, F.-J. Hambsch, E.V. Gai, Soo-Youl Oh, S.A. Badikov, T. Kawano, H.M. Hofmann, H. Vonach, S. Tagesen, International Evaluation of Neutron Cross Section Standards, Nuclear Data Sheets, Volume 110, Issue 12, (2009), Pages 3215-3324, ISSN 0090-3752, http://dx.doi.org/10.1016/j.nds.2009.11.001. (http://www.sciencedirect.com/science/article/pii/S0090375209001008)

China Nuclear Data Center (CNDC) and the China Nuclear Data Coordination Network (CNDCN), "CENDL-3.1 - The Chinese Evaluated Nuclear Data Library", https://www.oecd-nea.org/dbforms/data/eva/evatapes/cendl_31/ (2009).

M.B. Chadwick, P. Oblozinsky, M. Herman *et al.*, "ENDF/B-VII.0: Next Generation Evaluated Nuclear Data Library for Nuclear Science and Technology", Nucl. Data Sheets, 107, 2931-3060, (2006).

M.B. Chadwick, M. Herman, P. Oblozinsky, *et al.*, "ENDF/B-VII.1 nuclear data for science and technology: Cross sections, covariances, fission product yields and decay data", Nucl. Data Sheets, 112, 2887-2996 (2011).

Cross Section Evaluation Working Group (CSEWG) "A CSEWG Retrospective," ed. C. Dunford, Brookhaven National Laboratory Report BNL-52675 (2001).

D. Cullen, "PREPRO 2015", https://www-nds.iaea.org/public/endf/prepro/ (2015).

J.I. Márquez Damián, D.C. Malaspina, J.R. Granada, "Vibrational spectra of light and heavy water with application to neutron cross section calculations", J. Chem. Phys., Vol. 139, p. 024504 (2013).

DataCite, http://www.datacite.org (2015).

P. Descouvemont, D. Baye, Rep. Prog. Phys. **73**, 036301 (2010).

M.E. Dunn, N. M. Greene, "AMPX-2000: A Cross-Section Processing System for Generating Nuclear Data for Criticality Safety Applications," International Congress on Advanced Nuclear Power Plants (ICAPP) embedded topical meeting of the American Nuclear Society 2002 Annual Meeting, June 9-13, Hollywood FL, USA (2002); http://web.ornl.gov/sci/scale/overview/ampx.htm.

Evaluated Nuclear Structure Data File (ENSDF), http://www.nndc.bnl.gov/ensdf/ (2015).

Experimental Nuclear Reaction Data (EXFOR), https://www-nds.iaea.org/exfor/exfor.htm (2015).

P. Fröbrich, R. Lipperheide, *Theory of Nuclear Reactions*, Oxford University Press, ISBN: 0-19-853783-2 (1996).

F.H. Fröhner, "Evaluation and Analysis of Nuclear Resonance Data," JEFF Report 18 (2000).

H. Goldstein, "Panel Discussion", Proc. Conf. Neutron Cross-Sections and Technology, NBS-299 2 1309 (1968).

V.B. Gundersen, Z.W. Hendrikse, "BibTeX as XML markup", http://bibtexml.sourceforge.net/ (2015).

A. Gurbich, "Ion Beam Analysis Nuclear Data Library", https://www-nds.iaea.org/exfor/ibandl.htm, IAEA (2014).

B.D. Hehr, Ph.D. Thesis, North Carolina State University (2010).

"Table of Isotopes", by N. Holden, *The CRC handbook of Chemistry and Physics* (2004).

H.C. Honeck, "ENDF, Evaluated Nuclear Data File, Description and Specifications", BNL-8381, Brookhaven National Laboratory (January 1965).

H.C. Honeck,"ENDF/B - Specifications for an Evaluated Nuclear Data File for Reactor Applications", BNL-50066, Brookhaven National Laboratory (May 1966), revised by S. Pearlstein (July 1967).

R.J. Howerton, *et al.*, "OMEGA, A CRAY 1 Executive Code for LLNL Nuclear Data Libraries", Lawrence Livermore National Laboratory Report UCRL-50400, Vol. 25 (1983).

D. Hughes, J. Harvey, *Selected Reference Material on Atomic Energy, Vol. 5: Neutron Cross Sections*, Brookhaven National Laboratory Report BNL-325, (1955).

International Atomic Energy Agency Nuclear Data Services, https://www-nds.iaea.org (2015).

International Union of Pure and Applied Chemistry (IUPAC) www.iupac.org (2015).

JEFF Scientific Co-ordination Group, "JEFF-3.2 evaluated data library", https://www.oecd-nea.org/dbforms/data/eva/evatapes/jeff_32/ (2014).

G.R. Keepin, T. R. Wimett, and R. K. Zeigler, J. Nucl. Energy **6**, 1 (1957).

O. Klein and Y. Nishina, Z. Phys. 52, 853 (1929).

J.U. Koppel, D.H. Houston, "Reference Manual for ENDF Thermal Neutron Scattering Data", ENDF-269, Brookhaven National Laboratory (1978).

A.M. Lane, R.G. Thomas, Rev. Mod. Phys., **30**, pp 257-353 (1958).

N.M. Larson, "Updated users' guide for SAMMY: multilevel R-matrix fits to neutron data using Bayes' equations", Oak Ridge National Laboratory Report ORNL/TM-9179/R7, ENDF Report ENDF-364 (2006).

R.E. MacFarlane, D.W. Muir, "The NJOY Nuclear Data Processing System, Version-91", LA-12740-M, LANL, USA, (1994).

R.E. MacFarlane, A.C. Kahler, "Methods of Processing ENDF/B-VII with NJOY", Nuclear Data Sheets, Vol. 111, pp. 2739-2890, (2010).

R.E. MacFarlane, "The NJOY Nuclear Data Processing System, Version 2012", A.C. Kahler, Ed., LA-UR-12-27079, Los Alamos National Laboratory (2012).

C. Mahaux, H.A. Weidenmüller, *Shell-Model Approach To Nuclear Reactions*, North-Holland, Amsterdam (1969).

M. Mattes, J. Keinert, "Thermal Neutron Scattering Data for

the Moderator Materials H_2O, D_2O and ZrH_x in ENDF-6 Format and as ACE Library for MCNP(X) Codes", Report INDC(NDS)-0470, IAEA, Vienna, (2005).

C.M. Mattoon, B.R. Beck, N.R. Patel, N.C. Summers, G.W. Hedstrom, D.A. Brown, "Generalized Nuclear Data: A New Structure (with Supporting Infrastructure) for Handling Nuclear Data", Nuclear Data Sheets, Volume 113, Issue 12, December (2012), Pages 3145-3171, ISSN 0090-3752, http://dx.doi.org/10.1016/j.nds.2012.11.008 (http://www.sciencedirect.com/science/article/pii/S0090375212000944).

V. McLane, "ENDF/B-VI Summary Documentation, Supplement I; ENDF/HE-VI Summary Documentation", ENDF-201, 4th ed., BNL-NCS-17541, Brookhaven National Laboratory (1996).

S. Mughabghab, *Atlas of Neutron Resonances*, Elsevier, ISBN: 978-0-444-52035-7 (2006).

D. Neudecker, R. Frühwirth, H. Leeb, "Peelle's Pertinent Puzzle: A Fake Due to Improper Analysis", Nucl. Sci. and Eng., **170**, 54 - 60 (2012).

K.A. Olive *et al.* (Particle Data Group), "The Review of Particle Physics", Chin. Phys. C, 38, 090001 (2014); http://pdg.lbl.gov.

Office of Science and Technological Information (OSTI) http://www.osti.gov (2015).

K. Parker, "Physics of Fast and Intermediate Reactors", IAEA, Vienna, 3-11 August 1961 (1962).

K. Parker, Atomic Weapons Establishment, UK Report Number 0-70/63 (1963).

B. Pritychenko, E. Běták, M.A. Kellett, B. Singh, J. Totans, "The Nuclear Science References (NSR) database and Web Retrieval System," Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 640, Issue 1, (2011), Pages 213-218, ISSN 0168-9002, http://dx.doi.org/10.1016/j.nima.2011.03.018.

J. Pruet, D. Brown, B.R. Beck, D.P. McNabb, "Alternative approach to nuclear data representation", Nucl. Inst. Meth. B **244**, pp. 473–480 (2006).

C.W. Reich, M.S. Moore, Phys. Rev. **111**, p 929 (1958).

P. Ribon, J. M. Maillard, "Probability tables and Gauss quadrature: application to neutron cross sections in the unresolved energy range", CONF-860906-26 (Sep 1986), CEA-N-2485, (1986); P Ribon, "Resonance self-shielding calculation with regularized random ladders", Ann. Nucl. Energy (UK) 13:4, (1986); G Rimpault *et al.*, "Validation of new sub-group algorithms for resonance self-shielding in heterogeneous structures", INIS-XN-305, CONF-8906162, (1989); P Ribon, "Statistical probability tables CALENDF program", INIS-XN-305, CONF-8906162, (1989); C J Dean *et al.*, "Production of fine group data for the Ecco code", CONF-900418, (1990); A Hebert, M Coste, "Computing Moment-Based Probability Tables for Self-Shielding Calculations in Lattice Codes", NSE 142, 245-257, (2002).

G.R. Satchler, *Introduction To Nuclear Reactions*, John Wiley and Sons, ISBN: 0-470-26467-5 (1980).

J. Seco, F. Verhaegen, *Monte Carlo Techniques in Radiation Therapy*, CRC Press, ISBN 1466507926, 9781466507920 (2013).

K. Shibata, O. Iwamoto, T. Nakagawa, N. Iwamoto, A. Ichihara, S. Kunieda, S. Chiba, K. Furutaka, N. Otuka, T. Ohsawa, T. Murata, H. Matsunobu, A. Zukeran, S. Kamada, and J. Katakura: "JENDL-4.0: A New Library for Nuclear Science and Engineering," J. Nucl. Sci. Technol. 48(1), 1-30 (2011). http://wwwndc.jaea.go.jp/jendl/j40/j40.html

V.V. Sinitsa, A.A. Rineiski, V.V. Tebin, "Probabilistic Representation of Resonance Structure in Nuclear Safety Applications," OECD/NEA/Subgroup 38 Meeting, NEA Headquarters, Paris, France May 12-16 (2014).

Ed. by E.E. Stoller, K. Nordlund, S.P. Simakov, "Summary Report of the Technical Meeting on Primary Radiation Damage: from nuclear reaction to point defects", IAEA Headquarters, Vienna, Austria, 1–4 Oct. 2012, INDC(NDS)-0624 (2012)

J.S. Story, M.F. James, W.M.M. Kerr, K. Parker, I.C. Pull and P. Schofield, Proc. 3rd Int. Conf. Peaceful Uses of Atomic Energy, Geneva, 31 August - 9 September 1964 (1965).

R.G. Thomas, Phys. Rev. **97**, p 224 (1955).

I.J. Thompson, F.M. Nunes, *Nuclear Reactions for Astrophysics*, Cambridge Press, ISBN: 978-0-521-85635-5 (2009).

Ed. by A. Trkov, M. Herman, D.A. Brown, ENDF/B-VII.1 Manual, *ENDF-6 Formats Manual: Data Formats and Procedures for the Evaluated Nuclear Data Files ENDF/B-VI and ENDF/B-VII,* CSEWG Document ENDF-102, BNL Report BNL-90365-2009 Rev.2 (2011).

Unified Modeling Language (UML) Resource Page, http://www.uml.org/ (2015).

Virtual Atomic and Molecular Data Center (VAMDC) http://www.vamdc.org/ (2015).

L. Van Hove, Phys. Rev. 95, 249 (1954).

E.Dupont *et al*, Nucl. Data Sheets, **120**, 264 (2014); http://www.oecd-nea.org/science/wpec/.

WPEC Subgroup 38 (SG38), "Beyond the ENDF format: A modern nuclear database structure", http://www.oecd-nea.org/science/wpec/sg38/ (2012).

WPEC Subgroup 38 (SG38), "Requirements for a new nuclear data structure - Part 1: Vision and Goals", NEA/NSC/WPEC/DOC(2013)441, available at https://www.oecd-nea.org/science/docs/2013/nsc-wpec-doc2013-441.pdf (May 12, 2013)

WPEC Subgroup 38 (SG38), "Requirements for a new nuclear data structure - Part 2: Implementation Plan", NEA/NSC/WPEC/DOC(2014)450, available at https://www.oecd-nea.org/science/docs/2014/nsc-wpec-doc2014-450.pdf (May 12, 2013)

WPEC Subgroup 38, "Requirements and specifications for a particle database", in preparation (2015).

WPEC Subgroup 38, "General-Purpose Data Containers for Science and Engineering", in preparation (2015).

yEd Graph Editor, http://www.yworks.com/en/index.html (2015).

S.V. Zabrodskaya, A.V. Ignatyuk, V.N. Koshcheev, M.N. Nikolaev, *et al*, "RUSsian File Of evaluated Neutron Data (RUSFOND)," http://www.ippe.ru/podr/abbn/english/libr/rosfond.php (2010).

G. Žerovnik, A. Trkov, D.L. Smith, R. Capote, Nuclear Instruments and Methods in Physics Research A 727, pp 33-39 (2013).

List of Requirements

0. WPEC Subgroup 38 tasks, 3
1. Main, 4
2. An example, 5
3. Hierarchical structures, 9
4. Legacy data, 11
5. `<library>`, 12
6. `<evaluations>`, 12
7. Particles, 14
8. Documentation, 18
9. `<representations>`, 20
10. Linking between original and derived data, 21
11. `<functionDef>`, 24
12. General low-level, 25
13. List of general purpose elements, 26
14. Text elements, 27
15. Hyperlinks, 27
16. Placeholder hyperlinks, 27
17. `<evaluation>`, 29
18. `<metaEvaluation>`, 32
19. All reaction lists, especially `<reactions>`, 37
20. `<summedReactions>`, 37
21. `<reaction>`, 38
22. Reaction designation, 40
23. `<crossSection>`, 40
24. `<dcrossSection_dOmega>`, 41
25. `<dcrossSection_dOmega_dE>`, 41
26. `<reactionProducts>`, 41
27. `<orphanedProducts>`, 42
28. `<decayProducts>`, 42
29. `<product>`, 43
30. `<distributions>`, 44
31. `<multiplicity>`, 45
32. `<resonances>` element, 48
33. `<spinGroup>`, 49
34. `<channel>`, 49
35. Resolved resonance region (RRR), 51
36. Unresolved resonance region (URR), 52
37. Background R matrix, 53
38. The `<backgroundReaction>` element, 54
39. Where to put covariance data, 56
40. Cross–covariance linkage, 56
41. `<grid>` elements, 59
42. `<grid>` elements for continuous functions, 59
43. Multidimensional `<grid>` use, 60
44. `<covariance>` and `<correlation>`, 60
45. `<matrixData>`, 60
46. `<weightedSumOfCovariances>`, 61
47. `<matrixSandwich>`, 63
48. `<sensitivity>`, 63
49. Atomic reaction data, 67
50. Photo-atomic data, 68
51. Charged-particle elastic scattering, 71
52. Fission, 72
53. Fission Product Yields (FPYs), 74
54. Spallation, 77
55. Primary gammas, 78
56. Thermal scattering kernel, 80
57. Gaussian self-part scattering kernel, 81
58. Coherent Elastic Scattering, 82
59. Incoherent Elastic Scattering, 82
60. Short Collision Time Approximation, 83
61. Average kinetic energy and forward momentum, 85
62. Mubar, 85
63. CDFs, 86
64. URR probability tables, 86
65. Grouped transport data, 87
66. Bondarenko factors, 88
67. Production data, 88
68. DPA and KERMA, 89