Nuclear Energy Agency (NEA)
NEA/WKP(2026)4
September 2026
www.oecd-nea.org

NEA WORKING PAPER

Subgroup 50 Data-format Requirement Document for an Automatically Readable, Comprehensive and Curated Experimental Reaction Database MEDUSAL

A. LEWIS, D. NEUDECKER, A. KONING, et al.

OECD
BETTER POLICIES FOR BETTER LIVES

NEA
NUCLEAR ENERGY AGENCY

OECD NEA Working Papers describe preliminary results or research in progress by the author(s) and are published to stimulate discussion on a broad range of issues on which the OECD Nuclear Energy Agency works.

Comments on Working Papers are welcomed and may be sent to the Nuclear Energy Agency.

All Nuclear Energy Agency Working Papers are available at www.oecd-nea.org/pub.

This work is issued under the responsibility of the Secretary-General of the OECD and does not necessarily reflect the official views of OECD Member countries or its Nuclear Energy Agency. The opinions expressed and arguments employed are those of the author(s).

Please cite this working paper as:

NEA (2026), "Subgroup 50 Data-format Requirement Document for an Automatically Readable, Comprehensive and Curated Experimental Reaction Database MEDUSAL", OECD Publishing, Paris, https://doi.org/10.82155/hhgw-nf78.

© OECD 2026

Attribution 4.0 International (CC BY 4.0).

This work is made available under the Creative Commons Attribution 4.0 International licence. By using this work, you accept to be bound by the terms of this licence (https://creativecommons.org/licenses/by/4.0).

Attribution – you must cite the work.

Translations – you must cite the original work, identify changes to the original and add the following text: *In the event of any discrepancy between the original work and the translation, only the text of original work should be considered valid.*

Adaptations – you must cite the original work and add the following text: *This is an adaptation of an original work by the OECD. The opinions expressed and arguments employed in this adaptation should not be reported as representing the official views of the OECD or of its Member countries.*

Third-party material – the licence does not apply to third-party material in the work. If using such material, you are responsible for obtaining permission from the third party and for any claims of infringement.

You must not use the OECD logo, visual identity or cover image without express permission or suggest the OECD endorses your use of the work.

Any dispute arising under this licence shall be settled by arbitration in accordance with the Permanent Court of Arbitration (PCA) Arbitration Rules 2012. The seat of arbitration shall be Paris (France). The number of arbitrators shall be one.

Subgroup 50 Data-format Requirement Document for an Automatically Readable, Comprehensive and Curated Experimental Reaction Database MEDUSAL

WPEC SG-50

July 29, 2026

SG-50 Requirements

Acknowledgements

The members of the Nuclear Energy Agency (NEA), the Nuclear Science Committee and the Working Party on International Nuclear Data Evaluation Co-operation (WPEC), acknowledge the authors of the working paper, including the co-Chairs of SG-50 (*), the monitor of SG-50 (**), and the NEA Secretariat (†).

Name	Country	Establishment
Devin BARRY	US	Naval Nuclear Laboratory
Jesse BROWN	US	Oak Ridge National Laboratory
Allan CARLSON	US	National Institute of Standards and Technology
Yaron DANON	US	Rensselaer Polytechnic Institute
Michael FLEMING†		OECD Nuclear Energy Agency
Daniela FOLIGNO		OECD Nuclear Energy Agency
Anastasia GEORGIADOU		OECD Nuclear Energy Agency
Michael GROSSKOPF	US	Los Alamos National Laboratory
Robert HAIGHT	US	Los Alamos National Laboratory
Andrew HOLCOMB		OECD Nuclear Energy Agency
Keegan KELLY	US	Los Alamos National Laboratory
Arjan KONING**		International Atomic Energy Agency
Amanda LEWIS*	US	Naval Nuclear Laboratory
Denise NEUDECKER*	US	Los Alamos National Laboratory
Shin OKUMURA		International Atomic Energy Agency
Naohiko OTSUKA		International Atomic Energy Agency
Boris PRITYCHENKO	US	Brookhaven National Laboratory
Georg SCHNABEL		International Atomic Energy Agency
Henrik SJÖSTRAND	Sweden	Uppsala University
Donald SMITH	US	Argonne National Laboratory
Scott VANDER WIEL	US	Los Alamos National Laboratory
Noah WALTON	US	University of Tennessee, Knoxville
Jimin WANG	China	China Nuclear Data Center
Viktor ZERKIN	Austria	International Atomic Energy Agency

Foreword

This Working Paper constitutes the requirement document that guides the development of the experimental reaction database, Machine-readable Experimental Data User App & Library (MEDUSAL), created by the Nuclear Energy Agency's (NEA) Working Party on International Nuclear Data Evaluation Co-operation (WPEC), Subgroup 50. Experimental reaction data are usually stored in the Nuclear Reaction Physics Database (EXFOR) in EXchange FORmat (EXFOR format). With MEDUSAL, the WPEC SG-50 wants to go beyond the EXFOR format and database to generate a library that is (a) automatically readable, (b) comprehensive and (c) curated. To this end, a definition is provided here:

- How metadata and data should be described to be unambiguously interpretable by codes and algorithms;
- What information should be stored for each datatype;
- How the various stages of information and judgements on a dataset beyond the original publication can be preserved. This information can range from updating a monitor cross-section to expert judgement applied to a data set for a specific evaluation.

This database can serve as more automatically readable input for future evaluations, ML codes and preserves expert judgement of many man-years.

The codes have been applied to a few selected files of different observables to showcase what these newly formatted data look like.

The NEA Working Party on International Nuclear Data Evaluation Co-operation (WPEC) was created in 1989, under the aegis of the Nuclear Science Committee (NSC), to improve the quality and completeness of nuclear data by bringing together representatives of the major nuclear data evaluation projects of NEA member countries and of selected invitees.

Executive summary

Experimental nuclear reaction data underpin evaluated nuclear data used in areas as varied as criticality safety, neutron dosimetry, nuclear energy, non-proliferation and treaty verification. This Working Paper examines how these data are stored and used ,and identifies a gap between tradi--tional data compilation and the needs of modern, data intensive methods. It concludes that while the existing Nuclear Reaction Physics Database EXchange FORmat (EXFOR) remains essential, it is not sufficiently structured for systematic, automated use. This document therefore proposes a new approach that builds on EXFOR rather than replacing it, by reorganising and enriching the same data to improve consistency, transparency and usability across a wider range of applications.

The analysis starts from EXFOR as the main international repository of experimental nu--clear reaction data, containing tens of thousands of experiments and data sets drawn from published literature. EXFOR is designed for expert use, with information structured for human interpretation. The document identifies limitations that arise when users attempt to analyse large volumes of data or integrate them into computational workflows. These include inconsistent descriptions of experimental conditions, uncertainty information presented in varied or incomplete ways, and the absence of a systematic record of how data have been corrected or selected in later evaluations. The document translates these observations into a set of requirements based on practical use cases, covering access, data structure, and treatment of uncertainties and corrections. The central message is that existing data should be retained but reorganised. EXFOR remains the reference source for compiled experimental results, and this document does not recommend rebuilding or replacing it. Instead, it proposes a complementary system, a Machine-readable Experimental Data User App & Library (MEDUSAL), which would reorganise EXFOR data into a more structured and machine readable form. The aim is to make the same information easier to use in automated tools, large scale comparisons and reproducible workflows. In practical terms, this means shifting from a model designed primarily for reading and consultation towards one that supports consistent processing and reuse. The proposal therefore focusses on improving how data are structured, accessed and documented, rather than on creating new data.

A key feature of the proposed approach is a layered structure that distinguishes between different types of information. The original experimental data are preserved unchanged in a first layer. Additional layers then record different kinds of updates. Objective corrections, such as adjustments based on updated reference standards, are stored separately from subjective inputs, such as expert judgement or curated selections of data. This separation allows users to understand what has been modified, and why, without losing access to the original measurements. It also addresses a current limitation, where important evaluation work is often not systematically recorded. By storing these steps explicitly, the system aims to avoid repeated manual processing and to make accumulated expertise more accessible to future users.
This document highlights the need for consistent access to data across different user groups, including evaluators, experimentalists and analysts using automated methods. It proposes that data should be accessible through a system that allows both targeted searches and large scale extraction. To support this, the way data are described must be standardised. Each element of metadata, such as experimental conditions or measurement type, should have a clear and consistent identifier. Units should be harmonised so that values can be compared directly without conversion. Observables, defined as the quantities being measured, should be described using a

consistent physical basis. These changes are intended to reduce ambiguity and make it easier to combine or compare data from different sources. This document identifies several routine steps that users currently perform repeatedly, such as converting relative measurements into absolute values or adjusting results to reflect updated reference data. It proposes embedding these processes within the database itself, so that they are applied consistently and recorded for future use. The treatment of uncertainties is a particular focus. In many cases, published data do not include all relevant sources of uncertainty, which can affect their interpretation. The proposed system would identify missing components using standard templates and allow additional estimates to be introduced in a transparent way. It also anticipates the need to identify data that are inconsistent with others, so that users can take this into account during analysis. The objective is to reduce variability in how data are processed and improve the reliability of comparisons.

This Working Paper concludes that the current experimental nuclear data infrastructure should evolve rather than be replaced. EXFOR remains the authoritative source of compiled experimen--tal data, but its structure limits its use in modern analytical contexts. The proposed approach builds on this foundation by introducing a new system that standardises data description, sepa--rates original information from subsequent processing, and embeds routine analytical steps within the database. The overall recommendation is therefore to retain the existing database while developing a complementary architecture that makes the same data more systematic, traceable and suitable for large scale and automated use.

List of abbreviations and acronyms

AI	Artificial Intelligence
AIACHNE	AI/ML Informed cAlifornium CHi Nuclear data Experiment
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
DOE	US Department of Energy
DOI	Digital Object Identifier
EXFOR	EXchange FORmat (Nuclear Reaction Physics Database)
FAIR	Findable, Accessible, Interoperable, and Reproducible
GUI	Graphical User Interface
IAEA	International Atomic Energy Agency
JSON	JavaScript Object Notation
LANL	Los Alamos National Laboratory (United States)
MEDUSAL	Machine-readable Experimental Data User App & Library
ML	Machine Learning
NDS	Nuclear Data Section (IAEA)
NEA	Nuclear Energy Agency (OECD)
NRDC	Nuclear Reaction Data Centers
NSC	Nuclear Science Committee
NSR	Nuclear Science References
OECD	Organisation for Economic Co-operation and Development
PFNS	Prompt Fission Neutron Spectrum
REST API	Representational State Transfer API
RIPL	Reference Input Parameter Library
SQL	Structured Query Language
WPEC	Working Party on International Nuclear Data Evaluation Co-operation (NSC NEA)
XML	eXtensible Markup Language

Table of contents

Foreword 4

Executive summary 5

List of abbreviations and acronyms 7

1 Introduction 1
1.1 The different layers of the MEDUSAL database 2
1.2 Philosophy of the database 2

2 Use cases 4
2.1 Nuclear data evaluator 4
2.2 Experimentalist 5
2.3 Model development 6
2.4 Mining of data and metadata with machine learning and artificial intelligence . . 7
2.5 Assessing quality of nuclear data libraries by comparing to experimental data . . 8

3 High-level requirements on MEDUSAL given different use cases 10
3.1 Access of data 10
3.2 Format 10
3.3 Data treatment 11
3.4 Storing past judgements 13

4 Database hierarchy 15
4.1 Introduction 15
4.2 Cross-sections 15
4.2.1 Total cross-sections 17
4.2.2 Exclusive channel cross-sections 17
4.2.3 Discrete level cross-sections 18
4.2.4 Total particle production cross-sections 18
4.2.5 Residual production cross-sections 19
4.3 Angular distributions (single-differential) 19
4.3.1 Elastic scattering angular distribution 19
4.3.2 Discrete level scattering angular distributions 20
4.4 Prompt, delayed and total fission neutron multiplicities 20
4.5 Prompt and delayed fission neutron spectra 21
4.6 Fission yields 22
4.6.1 Primary fission fragment yield 22
4.6.2 Independent fission product yield 22
4.6.3 Cumulative fission product yield 23

5 Data categories 24
5.1 General containers 24
5.2 Observable 27
5.3 Bibliography 28

5.4 Incident particles . . . 29
5.5 Detectors . . . 30
5.6 Samples . . . 31
5.7 Response function . . . 32
5.8 Data sources . . . 33
5.9 Correlation to other data sets . . . 34

6 IAEA nuclear data section EXFOR tools and databases 35
6.1 Development of python EXFOR parsers . . . 35
6.1.1 EXFOR Master Files . . . 35
6.1.2 EXFOR format . . . 36
6.1.3 Summary of the EXFOR parsers functionalities . . . 37
6.1.4 Conversion to JSON and tabular formats . . . 38
6.1.5 Augmenting EXFOR data with information from other sources . . . 38
6.1.6 EXFOR dictionary in JSON . . . 39
6.1.7 REST APIs and web interfaces . . . 39
6.2 EXFOR dissemination systems . . . 39

7 Use case for employing SG-50 formatted data for machine learning 41
7.1 Background . . . 41
7.2 Creating the curated database . . . 42
7.3 Example from the curated database . . . 43
7.4 Input from ML for future curated databases . . . 48

8 Summary 49

9 References 51

1. Introduction

The aim of this document is to establish the requirements for an automatically readable, compre--hensive and curated experimental reaction database starting from EXFOR. These requirements will guide the development of the Machine-readable Experimental Data User App & Library (MEDUSAL database) by the Nuclear Energy Agency (NEA) Working Party on International Evaluation and Cooperation (WPEC), Subgroup 50 (SG-50).

The EXFOR database (Otuka et al., 2014) is the largest database of low- and intermediate-energy nuclear experimental reaction data. It encompasses more than 23 000 experiments with more than 177 000 data sets as of January 2022. The results and information about each data set is compiled as it was published (in consultation with authors where possible) unless obvious errors are found. These compiled data sets provide essential and easy-to-access input for nuclear data evaluations. Evaluations in turn serve as input to simulations that answer questions spanning areas from astrophysics, criticality safety, global security, neutron dosimetry, nuclear energy, to non-proliferation and treaty verification. Of course, using data in EXFOR for nuclear data evaluations is only one of many use cases, some of which are documented here.

The EXFOR database and format emerged from several stakeholders recognising the need for an easily readable exchange format of experimental reaction data for nuclear data evaluation. The first compilations of such data were undertaken at Los Alamos National Laboratory (LANL), the Metallurgical Laboratory, and University of Chicago from 1945 to 1947, and were then further developed at the National Nuclear Data Center of Brookhaven National Laboratory. In the time-frame of 1963–1964, additional data centers were established: the Nuclear Data Section of the International Atomic Energy Agency in Austria, the Nuclear Energy Agency in France, and Obninsk in the then USSR. These four data centers established the EXFOR exchange format after 1969. The Nuclear Reaction Data Centers (NRDC), a network of data centers co-ordinated by the International Atomic Energy Agency (IAEA), currently maintains and updates the EXFOR database, format, and tools (Okumura et al., 2024; V. V. Zerkin, 2022).

Recently, templates of expected measurement uncertainties were published (Lewis et al., 2023; Lewis, Amanda M. et al., 2023; Neudecker, Carlson, Croft, et al., 2023; Neudecker, Devlin, Haight, et al., 2023; Neudecker, Lewis, Matthews, et al., 2023; Neudecker et al., 2018; Vanhoy et al., 2023). These templates list what uncertainty and measurement information should be provided for traditional measurement types of fission, total, capture, charged particles, and scattering cross-sections as well as prompt fission neutron spectra or neutron multiplicities. These templates also provide stand-in values that can be used to estimate missing uncertainty values for past experiments. Lists of expected measurement uncertainties can be used in WPEC SG-50 codes to flag which uncertainty sources are likely missing in particular EXFOR entries. The template uncertainty values can be also used to automatically augment these missing uncertainty sources based on a published source to help evaluators automatise part of the uncertainty quantification.

The value of EXFOR cannot be overstated. It makes many datasets openly available with metadata information that might be otherwise hard to come by. However, there are some aspects of the EXFOR database and format that impede the most effective use of EXFOR data for nuclear data evaluations:

- The EXFOR format allows for ambiguities in interpreting metadata and data, which limit automated reading of EXFOR files for larger-scale nuclear-data evaluations or analysing large amounts of data with modern algorithms (e.g. machine learning ones). For instance, the uncertainties are described with uninformative labels such as "ERR-x" with free-text descriptions based on the original work of the author,
- Information in the EXFOR database is compiled from the original author's work and/or authorised by the authors. Subjective or objective judgements passed by evaluators on data and uncertainties are not captured by EXFOR. Therefore, evaluators have to reanalyse historic data and estimate total covariances for every new evaluation. This is a time-intensive undertaking and poor use of time for a limited number of nuclear data evaluators. The EXFOR correction system (V. V. Zerkin, 2022) on top of the EXFOR database addresses some of these issues (renormalising data with updated standards, monitors or structure values, adding/changing uncertainties, etc.).

The goal of NEA/WPEC SG-50 is to address these issues. To this end, this requirement document showcases how a format and database can be developed that:

- stores both the metadata and results of experimental reaction data sets such that they can be automatically read and unambiguously interpreted;
- enables preserving knowledge of experts beyond the authors of a dataset for future evalua--tions.

Chapter 2 defines the use cases focussed on for this work, and chapter 3 describes the high-level requirements on the database for each use case. Chapter 4 describes the hierarchy and observable categories planned for MEDUSAL and compares them to those in EXFOR. Chapter 5 defines metadata categories such that they meet these criteria. Chapter 6 describes the EXFOR codes and databases developed and maintained by the IAEA, and an application example is given in chapter 7.

1.1. The different layers of the MEDUSAL database

The MEDUSAL database will have several layers. These layers make it clear for the user what data are from the original EXFOR entry, or whether objective or subjective corrections were stored.

- Layer 0: translation of EXFOR-formatted files into new format, no changes to data;
- Layer 1: added information that is in the literature but not in EXFOR;
- Layer 2: objective corrections (new monitors coming from the Neutron Data Standards, decay data), listing of missing uncertainties compared to those listed for templates of expected measurement uncertainties, and objective testing requirements (p-values, missing corrections);
- Layer 3: subjective corrections, uncertainties corrected by adding uncertainties from templates of expected measurement uncertainties, expert judgement containers, which data were used for a particular evaluation.

1.2. Philosophy of the database

- Keep close to the existing EXFOR format and reuse what is already working well. This approach is taken with the hope that some additions in the WPEC SG-50 requirement document compared to the existing EXFOR format might be considered for adoption in

the future by the NRDC.

- Keep metadata separate from but linked to the datasets, in a setup similar to the EXFOR subentries, which do not contain redundant metadata.
- Deviate from EXFOR format insofar as there will be no work on the level of sub-entries. Instead, a dataset will be identified by EXFOR number, accession number (sub-entry) and dataset pointer. This decision was taken because sometimes several datasets are stored in one sub-entry that can stay on their own from an evaluation perspective, i.e. all information needed for an evaluation is in this one entry. Exceptions will be made where several datasets together are needed for complete information for evaluation purposes. Thus, a unique identifier will exist per dataset with length 9, while in EXFOR format it has length 8.

2. Use cases

The EXFOR database strives to document and preserve experimental nuclear reaction data true to its original publication. MEDUSAL's goal, on the contrary, is to provide the best possible experimental input to support nuclear-data evaluations. To accomplish this, the database must be easy to read, comprehensive and must be able to preserve information on data sets gained through previous evaluations. As MEDUSAL is geared towards the use of experimental nuclear reaction data, rather than preservation of the originally reported data, it is important to understand the different use cases that the library must cater to.

Typical users and use cases are:

- nuclear data evaluators who produce the best average result from all existing data sets including uncertainty quantification;
- experimentalists who compare to previous measurements and gain an understanding of similarities between aspects of past measurements;
- developers and users of nuclear model codes who either need experimental data as input for their models, or for comparison to their model predictions;
- machine learning or artificial intelligence algorithms which look for trends in data and metadata to better understand the physics represented by the experiments;
- users who assess the quality of nuclear data libraries by efficient comparison with experi--mental data.

These different use cases, their goals and needs regarding MEDUSAL are summarised below.

2.1. Nuclear data evaluator

Goal of the user: The goal of this user is to evaluate one or more nuclear-data observables including the best understanding of experimental data and theory at the time. To this end, they will use available and trustworthy experimental data as direct input to the evaluation. Auxiliary experimental data might also be used to fit or guide model parameters[1].

High-level requirements

Use of retrieved data: The evaluator will use metadata, data and past judgements of the observable of interest to:

- judge the quality of the data and either accept or reject them for the evaluation;
- re-normalise data according to the newest monitor reactions;
- estimate experimental uncertainties and add missing uncertainties;
- estimate correlations for uncertainties of the same or between different data sets;
- identify outliers and try to reconcile discrepant data sets;

[1]For instance, experimental data for post-neutron emission fission yields as a function of mass can be used to guide the parameterisation of pre-neutron emission fission yields needed as input to CGMF (Talou et al., 2021) or FREYA (Verbeke et al., 2018) calculation. CGMF and FREYA model predictions, in turn, can be merged with average prompt neutron multiplicities or prompt-fission neutron spectra to give evaluated data for the latter two observables. Post-neutron emission fission yields, average prompt-fission neutron multiplicities and prompt-fission neutron spectra will all be stored in MEDUSAL.

- truncate and correct data according to their expert judgement;
- build a database based on all the available data and feed it into their evaluation.

Needs on accessing the data: Evaluators download all the data available for the observable(s) of interest. They would need all information stored in MEDUSAL entries and available in the literature, i.e. metadata, data, uncertainties and judgements on the data. The evaluator would need an Application Programming Interface (API) to download and search the data. A plotting API with selected data would also be helpful.

Examples of typical queries to the database could be:

- download all ^{237}Np(n,f) cross-sections;
- download all double differential (p,xn) spectra for all nuclides for 10–50 MeV;
- download all experiments with detector type A, neutron-flux type B, etc.;
- search by DOI links;
- search by selected keywords;
- search by curated data;
- download all data that were used for the ENDF/B-VIII.0 evaluation of ^{56}Fe(n,tot) cross-section;
- search by use in multiple evaluations.

What layers of the database will be accessed? All layers of the database will be accessed.

What information will be fed back into the database?

- missing information that they found in the literature but not in MEDUSAL (uncertainties, metadata, etc.);
- expert judgement on a dataset such as unreasonable uncertainties, possible biases, etc.;
- total covariances and data as used for their evaluation;
- retroactive quantification of judgements from evaluations or experimental studies.

Benefit of Using MEDUSAL for this User: The nuclear data evaluator would save time thanks to MEDUSAL because they:

- can quickly retrieve data;
- obtain data that is automatically renormalised by newest monitors;
- get data with uncertainties and correlations enhanced by templates of expected measurement uncertainties;
- can compare metadata of two experiments more easily due to unique identifiers;
- have one central storage place of past judgements on the data.

2.2. Experimentalist

Goal of the user: The goal of an experimentalist is to provide the best possible measurement of an observable at a time. Sometimes, they might want to achieve the highest precision possible, shed light on discrepancies between past measurements, or measure an observable in a specific energy range for the first time.

Use of retrieved data: Experimentalists use experimental data in MEDUSAL to justify the need for their experiment, find out how experiments were previously undertaken and compare to

historic data. To that end, experimentalists will use metadata, data and past judgements of the observable of interest to:

- find out how many measurements of an observable exist;
- research what analysis techniques and hardware were employed in past measurements to guide their own measurement or avoid known pitfalls;
- research whether past measurements were indicated to be biased or questioned;
- look at total uncertainties to understand what the state-of-the-art is they compete with;
- download curated data (outliers removed) with total uncertainties for plotting;
- compare their experimental data to past measurements.

Needs on accessing the data: Experimentalists download and look at all the data available for the observable(s) of interest. They would need metadata, data, total uncertainties and judgements on the data. A download and plotting API with selected data would be helpful. They would also perform API searches according to a clearly defined observable or by metadata.

What layers of the database will be accessed? Experimentalists would need access to the third layer of the MEDUSAL database.

What information will be fed back into the database?

- Expert judgement on a dataset such as possible biases.
- New experimental data could be uploaded in the database to plot the new data set in comparison to existing data. However, it should be emphasised that final new experimental data should be delivered through the NRDC to EXFOR first and, then, it will enter MEDUSAL through EXFOR.

Benefit of Using MEDUSAL for this User: The experimentalists would benefit from MEDUSAL because they can:

- compare their new measurement easily to curated experimental data;
- quickly assess the precision achieved and hardware/ analysis techniques used in past measurement;
- gain an understanding what data were identified as biased and if some aspects of a measurement correlates to data being biased.

2.3. Model development

Goal of the user: The goal of a model developer is to develop their model to predict available experimental data accurately. The goal of a model user is to fit model parameters or parameter systematics of a trusted model such that the model values predict experimental data to its best ability.

Use of retrieved data: A model developer or user will use large amounts of the MEDUSAL database, that is across many nuclides and observables, at once to develop models and parameter systematics by:

- comparing model-predicted quantities to curated experimental data that are deemed reliable, or

- by using trustworthy experimental data with complete uncertainties as direct input for their model.

Needs on accessing the data: The model developer or user will download large parts or all experimental data at once. They would retrieve:

- curated experimental data (i.e. outliers removed, renormalised to the newest monitor);
- realistic total uncertainties that were updated according to the templates of expected measurement uncertainties;
- data that have been used for a specific evaluation;

make this retrieval easier, a well-defined, automatically readable format (or codes for reading) with unique identifiers for data types would be needed.

What layers of the database will be accessed? Model developers or users would need access to the third layer of the MEDUSAL database.

What information will be fed back into the database? The model developer or users could feed back into MEDUSAL outliers that were automatically identified in a large number of data sets compared to theory. Model developers or users can also identify where experimental data are needed for better model predictions. While the latter information cannot directly feed back into MEDUSAL, knowing where the gaps are in historic data can motivate new measurements which in turn, end up in MEDUSAL through EXFOR.

Benefit of Using MEDUSAL for this User: The model developer or user would save time if using MEDUSAL because:

- Data are already curated. This feature reduces the need for rejecting outliers, renormalising data and adding missing uncertainties.
- They can quickly access and read large amounts of data that are already formatted for such large-scale reading.

2.4. Mining of data and metadata with machine learning and artificial intelligence

Goal of the user: Machine learning (ML) and artificial intelligence (AI) are often applied to large amounts of data with the goals to:

1. Find trends in data and metadata to better understand the physics represented by the experiments (Whewell et al., 2020),
2. Inform models with the data (Lovell et al., 2022).
3. Find issues in nuclear-data libraries (Neudecker et al., 2021).

The needs to meet the second goal coincide with those described for the user "Model developer and user", while the third goal aligns with those needs outlined for the user described in the next section. Here, the main focus lies on the needs to meet the first goal.

Use of retrieved data: To meet the first goal, experimental data, uncertainties, and metadata would be downloaded for several observables at a time to perform the following:

- identify outliers in experimental data;

- correlate data being outliers with specific metadata of the experiments.

Needs on accessing the data: This user will download large parts or all experimental data at once. They would retrieve curated experimental data, total uncertainties augmented by the templates, partial uncertainties, flags identifying possible outliers/ biases, metadata, and comments from previous users. In order to undertake an ML/AI analysis, the data first need to be renormalised to the newest monitors and missing uncertainties must be added. Also, a well-defined, automatically readable format is needed to quickly download a large amount of data. A unique format for all metadata is key to enabling this type of analysis.

What layers of the database will be accessed? All layers of the database will be accessed.

What information will be fed back into the database? One can feed back into MEDUSAL via ML and AI methods the following information for a large amount of data automatically:

- Which data are outliers in a large number of data sets?
- What metadata features could be possibly related to data points being outliers?

ML/AI can also motivate where experimental data are needed to either fill gaps in the database or inform evaluations. In addition to that, they can point towards what features of past measurements led to data being outliers and what features should therefore be avoided in new measurements or explored as to why they are correlated with biases in data.

Benefit of Using MEDUSAL for this User: MEDUSAL could enable ML/AI to be applied to find trends in large amounts of data by providing:

- easy access to large amounts of data;
- metadata in a unique and easy to interpret format;
- curated data that are renormalised and encompass all pertinent uncertainties.

2.5. Assessing quality of nuclear data libraries by comparing to experimental data

Goal of the user: The user wants to understand if data from a nuclear data library are realistic given differential experimental data.

Use of retrieved data: Curated experimental data and uncertainties from MEDUSAL will be used to compare to evaluated nuclear data from a library. This can be useful for determining where evaluation and experimental effort should be focussed. Discrepancies, or a lack of data for comparison, are important for applications to understand.

Needs on accessing the data: This user will plot curated experimental data and total uncertainties enhanced by the template via an API for one reaction at a time and compare to evaluated data. It would be helpful for this user to be able to retrieve those data that have been used for a specific evaluation.

What layers of the database will be accessed? This user would need access to the third layer of the MEDUSAL database.

What information will be fed back into the database? This user would likely flag outlying data by eye.

Benefit of MEDUSAL for User This user would benefit from MEDUSAL by:

- having data in an easy-to-parse format;
- having curated data and uncertainties, which will save the user time on renormalising data, removing outliers and adding uncertainties.

One could go as far as to say that this use case is enabled only by the availability of curated experimental data. If data are only used as presented in a journal article (normalised to old monitor, without all pertinent uncertainties), a direct comparison of these data to evaluated nuclear data might be misleading. This is, of course, also a stumbling block for the use cases of model development and mining of MEDUSAL with ML/AI.

3. High-level requirements on MEDUSAL given different use cases

High-level requirements on the MEDUSAL database, codes and format are formulated below based on the different use cases documented in Section 2. Each high-level requirement is described concisely, and a short description is provided on how parts of MEDUSAL will address these requirements.

3.1. Access of data

Enabling download of a large amount of data at once

Users that develop models or apply ML/AI algorithms to large amounts of data in MEDUSAL would benefit from a capability to download large parts of the MEDUSAL database at once. For instance, experimental data across many nuclides and observables could be sought at the same time for all layers of MEDUSAL. All information stored for a dataset might be needed, i.e. experimental data, uncertainties and metadata.

A MEDUSAL API is planned that enables download of a large amount of data at once.

API to access and plot the data of all different layers

Nuclear data evaluators, experimentalists and validators of nuclear data libraries could want an API to download experimental data, uncertainties and metadata from MEDUSAL for one specific isotope and observable at a time. Experimentalists might also choose to download data that used specific metadata features (e.g. a specific neutron-producing reaction, detector type). Again, all layers of the database including metadata, data and uncertainties should be accessible by this API. Plotting these data would also be of interest.

An API is planned as part of the MEDUSAL code suite to support this undertaking. This API would have a user-interface where one can select data from a list.

Uploading user-defined data for plotting to compare to data in MEDUSAL

Experimentalists and evaluators might want to upload user-defined data (e.g. their new experimental data or evaluation) for plotting their data against the MEDUSAL database. This might become a feature of the MEDUSAL code suite.

3.2. Format

Creating a format that is easy to read automatically for a large amount of data

All users would need to be able to download the data in a format that is easy to parse. In particular, large-scale users (for model development or applying ML/AI) need a format that is easy to read automatically.

This high-level requirement leads to several detailed requirements on a format for MEDUSAL which is described in Section 5 of this document. The specification document (SG-50, 2024) for the MEDSUAL format will set forth how these detailed requirements are being addressed.

Clear identification of what observable was measured and how it relates to ENDF-6 formatted nuclear-data observables

A clear identification of the measured observable is the starting point for every user. If the measured observables are not clearly identified, users are more likely to miss part of the experiments available for their purposes. One way (but not the only one) to have a clear identification of observables is to describe it via MF/MT number of the ENDF-6 format. This would aid evaluators and validators of nuclear-data libraries in particular in identifying all relevant data. However, the link to the ENDF-6 observable identification will only be an auxiliary way to describe an observable, as other users might not be familiar with the MF/ MT formulation of observables in ENDF-6 format and not all observables stored in MEDUSAL will have an equivalent to an MF/ MT number of ENDF-6. Hence, a physics-based format should be used as the primary identification of an observable.

MEDUSAL addresses this requirement by clearly defining the categories of data encountered in MEDUSAL in Section 4. The philosophy of MEDUSAL is that each data set within the database will correspond to a single observable.

Using common units for all data in the database to make them easily comparable

In order to facilitate automatic reading and plotting of MEDUSAL data, all data should be reported in the same units. The reason for this requirement lies in the fact that downloading several data sets from EXFOR usually requires converting data into common units, a mundane task that can take time, and is error-prone if done for each individual data set.

Data will be converted into common units via the MEDUSAL code that converts data from layer 0 to 1. The common units to be used will be defined in the MEDUSAL specification document.

Using unique identifiers for metadata to easily find common features among experiments

It is key for users wanting to apply ML/AI to metadata to have them in a common format and describe them with unique identifiers. That means that one and the same physics process, uncertainty source, equipment, etc., should be described with only one unique word. For instance, one might find several terms to describe detector-efficiency uncertainties in EXFOR ("extrapolation to zero", "efficiency", "detector efficiency", "detector calibration"). While a human might be able to easily identify those as belonging to the all-encompassing term of "detector efficiency uncertainty", a computer cannot easily make that connection. Having one common identifier to describe the uncertainty category, metadata feature, etc., increases interpretability. It also helps users such as evaluators to quickly compare two data sets.

Section 5 of this document lays out which metadata need to be defined to adequately describe a measurement. The specification document outlines how the different metadata can be uniquely identified (i.e. a common language) and stored.

3.3. Data treatment

Converting ratio data to absolute data

Data in EXFOR are sometimes reported as ratios, e.g. ^{237}Np(n,f)/^{235}U(n,f) cross-sections. However, several users (evaluators, experimentalists, validators of nuclear-data libraries, etc.) would often be interested in comparing ^{237}Np(n,f) cross-sections of several experiments. These absolute cross-sections can be obtained by multiplying ratio data with reference cross-sections. This is another mundane data processing task that can take considerable time and can lead to errors.

Data in layer 2 of the MEDUSAL database will be converted to absolute data using the latest monitor data to render them comparable across several data sets. In order to achieve this, a clear format is needed for identifying monitor reactions as outlined in Section 5. This format is defined in the specification document.

Re-normalising to new standard and reference reactions

Data in EXFOR are presented true to the original publication. If ratio data were reported multiplied with a monitor of the time of the experiment, these data will be reported along with the monitor cross-section used (if available). If several data sets are studied at the same time, it is very likely that different monitors were used for different data sets to convert ratio to absolute data. Hence, data might seem systematically off-set with respect to each other. But part or all of the off-set could stem from different monitor data being used for one and the same observable.

Hence, one requirement on MEDUSAL is to always use one and the same monitor for the same reaction. In order to achieve this, a clear format is needed for identifying monitor reactions as outlined in Section 5. This format is defined in the specification document.

Automatically flagging and identifying outliers with various algorithms

One crucial step undertaken by evaluators and experimentalists is identifying outlying data sets. However, model developers, users applying ML/AI to MEDUSAL data, and validators of nuclear data libraries also benefit from having outliers either identified or removed so they won't bias their analysis. Hence, it would be helpful for all users of MEDUSAL if outliers would be automatically flagged.

Work is currently ongoing to identify adequate algorithms to identify outliers in MEDUSAL data. If adequate algorithms can be found, they might be implemented in MEDUSAL code suites. Data will be flagged as outliers in layer 2 of the MEDUSAL database. Identifying outliers in data triggers itself other high-level requirements. For instance, data need to be in an easy-to-read format, re-normalised to the newest standard, in common units and uniquely identified to apply algorithms to identify outliers.

Automatically flagging missing or suspiciously low uncertainties via templates

Data in EXFOR are stored as reported in their original publication or by the author. Uncertainties are also reported as presented by the author. In some cases, uncertainties reported might not encompass all sources expected for a specific measurement type. Usually, an evaluator would add missing uncertainties, in this case for complete uncertainty budgets. Otherwise, a data set with under-reported uncertainties might have an unreasonably large impact on the evaluation, which will be propagated to transport simulations (Neudecker et al., 2016, 2020). The results of other users (model developers, users applying ML/AI to MEDUSAL data) might also be inadequately biased by underestimated uncertainties. Hence, one requirement is to flag missing uncertainties.

Identifying missing uncertainties leads to the following three requirements:

- A list of uncertainties expected for a typical measurement type.
- The measurement type of an experiment needs to be clearly identified.
- The uncertainty sources in MEDUSAL must have a unique naming convention to easily identify which uncertainties are provided for a data set.

These requirements can be addressed by:

- Using templates of expected measurement uncertainties to identify missing uncertainty sources (Neudecker et al., 2020).
- The measurement type of an experiment can be identified clearly by metadata descriptors. Detailed requirements on the metadata are discussed in Section 5 of this document and the specification document.
- The uncertainty sources in MEDUSAL format will have a unique naming convention as described in the specification document.

The last step is then to identify missing uncertainty sources by the MEDUSAL code that converts data from layer 1 to layer 2.

Automatically adding missing uncertainties via templates

As mentioned above, several users would benefit from having curated experimental uncertainties at their disposal. Curated means that missing uncertainty sources are identified via templates of expected measurement uncertainties and then added to already reported uncertainties of layer 0 to generate the uncertainties in layer 3 of the database.

To this end, uncertainties proposed in the following references (Lewis et al., 2023; Lewis, Amanda M. et al., 2023; Neudecker, Carlson, Croft, et al., 2023; Neudecker, Devlin, Haight, et al., 2023; Neudecker, Lewis, Matthews, et al., 2023; Neudecker et al., 2018; Vanhoy et al., 2023) will be added in layer 3 via the MEDUSAL code suite for those uncertainty sources that were not reported in the literatureand are reported as missing in layer 2.

Estimating total covariances using existing uncertainties and templates

All users of the database would need total curated experimental uncertainties for their needs, either to compare data to each other or as direct input for their studies. However, uncertainties in EXFOR are reported as given by their author and might encompass all pertinent uncertainty sources or only a sub-set. This renders a direct comparison of data difficult.

MEDUSAL will build upon including missing uncertainties via templates to generate curated total uncertainties for layer 3. Total covariances will be also reported based on the following references (Lewis et al., 2023; Lewis, Amanda M. et al., 2023; Neudecker, Carlson, Croft, et al., 2023; Neudecker, Devlin, Haight, et al., 2023; Neudecker, Lewis, Matthews, et al., 2023; Neudecker et al., 2018; Vanhoy et al., 2023).

3.4. Storing past judgements

Storing past judgements on the data

Evaluators and experimentalists gain knowledge about past data (e.g. on missing corrections, uncertainties, etc.) through detailed studies of their own. This information is often lost as there is no central place to store this information. However, it could be very helpful to future users of the data to know what earlier users of data uncovered about the data.

MEDUSAL strives to store these past judgements in layer 3. A format needs to be developed to store this information and an API to save and upload the information.

Identifying if a data set was used for an evaluation

It might be helpful for some users (validators, evaluators and experimentalists) to know if a

specific data set was used for an evaluation as this lends credibility to a data set. It would also render evaluations more reproducible if the data could be stored as used for the evaluation.

The MEDUSAL database will store data used for the evaluation in layer 3 and introduce a flag that shows if data were used for particular evaluations.

4. Database hierarchy

4.1. Introduction

To have a proper categorisation of all the nuclear reaction observables for this database, it is best to take an independent view from the categorisation and nomenclature of EXFOR. Instead, for MEDUSAL the point of view of a "typical user" (defined in chapter 2) who wants to have direct access to experimental data has been taken. It was mentioned in chapter 3 that these users would require a clear identification of what observable was measured and how it relates to ENDF-6 formatted nuclear-data observables. Hence, they want to be shielded from the particular procedures which were used to build the EXFOR database, although all metadata reported in EXFOR, is of course, of vital importance and should be included in the new database, some of it reformatted or unified. Hence, instead of the categorisation of experimental data into so-called EXFOR "entries" and "subentries", the data will be categorised into "data sets". The main differences are that an EXFOR subentry may contain several unsorted data sets and that certain EXFOR reaction codes, and thus subentries, may have no equivalent in nuclear data libraries or theoretical model codes. In MEDUSAL, each data set will have a single observable, and one data set is typically what you find as one group of data with a common symbol in a plot, often compared with other data sets, nuclear model code predictions or nuclear data libraries. The observable categorisation in MEDUSAL will supplement the more experimental-method based categorisation of observables in EXFOR, and connections between the categories will be explored and documented. This may additionally help to clarify the myriad "types" of experimental data that can be useful in the evaluation of a physical reaction category, such as partial gamma intensities or capture yield data sets for capture cross-sections[2]. The idea is that bit by bit, the entire EXFOR database is mined for relevant data, leaving less relevant data for later (or never). The classification below is for the same observables one can find in an ENDF nuclear data library or as the output of a nuclear mode code like TALYS (Koning & Rochman, 2012). The examples are given for incident neutrons but may hold for any projectile.

4.2. Cross-sections

There are various types of cross-sections, and for the database one may distinguish 5 categories, namely total, exclusive channel, discrete level, total particle production and residual production cross-sections. On top of that, exclusive channel and residual production cross-sections each may have subcategories as they may have been measured for the final ground state or isomer separately. Table 4.1 gives the most important cross-sections. For nuclear applications, and also in some derived EXFOR databases like C4, these are often categorised by MT number. Because of that historic importance, when an official MT number is available, it will be retained as metadata in the MEDUSAL database, though they will no longer be the leading identifier as there are too many reactions without a proper MT number, and the MEDUSAL database is targeted towards to as many nuclear physicists as possible. As there are many TALYS users, the corresponding TALYS output file has been given as well.

[2]Currently, the MEDUSAL categories have not yet been fully fleshed out, but the start of the connection to the EXFOR reaction string and quantity codes are presented in the specifications document Sections 2.2 and 2.10-2.14. The 9^{th} quantity in the reaction string is not used here, as only experimental data is considered for MEDUSAL at this time.

Table 4.1: Most important cross-sections

Reaction	MT	TALYS output file
(n,tot)	1	totalxs.tot
(n,el)	2	elastic.tot
(n,non)	3	nonelastic.tot
total (n,n')	4	xs100000.tot
(n,n'$_1$)	51	nn.L01
(n,n'$_2$)	52	nn.L02
...	53-90	nn.L03....
(n,2n)	16	xs200000.tot
(n,3n)	17	xs300000.tot
(n,f)	18	fission.tot
(n,nα)	22	xs100001.tot
(n,np)	28	xs110000.tot
(n,γ)	102	xs000000.tot
total (n,p)	103	xs010000.tot
(n,p$_0$)	600	np.L00
(n,p$_1$)	601	np.L01
...	602-648	np.L02....
total (n,α)	107	xs000001.tot
(n,α_0)	800	na.L00
(n,α_1)	801	na.L01
...	802-848	na.L02....
(n,xn)	201	nprod.tot
(n,xp)	203	pprod.tot
(n,xα)	207	aprod.tot
(n,x)		rpZZZAAA.tot

Note: In Table 4.1, for historic reasons, the MT number is also given when available. A similar set exists for projectiles other than neutrons.

Total cross-sections

Definition of one data set

One data set for total cross-sections is defined by:

- the projectile;
- the target nucleus;
- the reaction type:
  - (n,tot)
  - (n,el)
  - (n,non);
- one or more incident energies with their uncertainties;
- cross-sections with their uncertainties,
- all other metadata.

Relation to EXFOR

This is probably the easiest category, in that often (but certainly not always) there is a one-to-one correspondence to our data set and what is stored in one subentry. Hence, for simple reactions like (n,tot), (n,el), (n,non) one data set often corresponds to one EXFOR subentry. The notable exceptions which require some more complex parsing are in the other reaction classes.

Exclusive channel cross-sections

Definition of one data set

One data set for exclusive channel cross-sections is defined by:

- the projectile;
- the target nucleus;
- the reaction type, e.g.:
  - (n,n')
  - (n,2n)
  - (n,f)
  - (n,g);
- optional: the residual ground state or isomer;
- one or more incident energies with their uncertainties;
- cross-sections with their uncertainties;
- all other metadata.

Relation to EXFOR

This is also a relatively easy category, in that often (but certainly not always) there is a one-to-one correspondence to our data set and what is stored in one subentry. Hence, for simple reactions like (n,n'), (n,2n), (n,g) one data set often corresponds to one EXFOR subentry. The notable exceptions which require some more complex parsing are in the other reaction classes. In principle, the state of the residual nuclide may be denoted in the EXFOR reaction code, though there may be some disagreement between the theoretical and EXFOR definition of the 'm' and 'n' isomer.

Discrete level cross-sections

Discrete level cross-sections are for reactions such as (n,n'$_1$), (n,n'$_2$),....(n,p$_0$), (n,p$_1$), etc.

Definition of one data set

One data set for discrete level cross-sections is defined by:

- the projectile;
- the target nucleus;
- the outgoing particle and level e.g.:
  - (n,p$_0$)
  - (n,a$_1$);
- one or more incident energies with their uncertainties;
- cross-sections with their uncertainties;
- all other metadata.

Relation to EXFOR

Discrete level cross-sections come with at least two complications:

(1) While most users are probably interested in e.g. the inelastic cross-section for the second discrete level of ^{56}Fe, in EXFOR the excitation energy has been coded, often because that is what was given in the paper. This means that this particular reaction may be present throughout the EXFOR database with slightly different excitation energies, while they all belong to the same reaction. To mine data out of EXFOR for our database, one needs to use a discrete level database, such as the one from RIPL (based on ENSDF) to identify all cases which belong to the same reaction.

(2) The other complication is that cross-sections for several discrete levels may be stored in one subentry in EXFOR, and these may be grouped per incident energy or per discrete level. Sorting codes for X4 to C4 exist, so one could choose for one consistent grouping, but in general this becomes complicated for secondary distributions with more running variables, e.g. incident energy, excitation energy, outgoing angle.

Total particle production cross-sections

These are usually denoted as (p,xn), (n,xa), etc. and are mostly for high incident energies.

Definition of one data set

One data set for total particle production cross-sections is defined by:

- the projectile;
- the target nucleus;
- the outgoing particle e.g.
  - (n,xp)
  - (p,xn);
- one or more incident energies with their uncertainties;
- cross-sections with their uncertainties;
- all other metadata.

Residual production cross-sections

These are usually denoted as (n,x), (p,x), etc. and are mostly for high incident energies, such as e.g. ^{56}Fe(p,x)^{49}Ti.

Definition of one data set

One data set for residual cross-sections is defined by:

- the projectile;
- the target nucleus;
- the final nucleus;
- optional: the residual ground state or isomer;
- one or more incident energies with their uncertainties;
- cross-sections with their uncertainties;
- all other metadata.

Relation to EXFOR

There may be a complication similar to the discrete level cross-sections. Sometimes, one has one EXFOR subentry per data set and the conversion is simple, but often several residual products are stored in one subentry and one has to do some awkward programming to identify the data points for the right data set.

In principle, the state of the residual nuclide may be denoted in the EXFOR reaction code, though there may be some disagreement between the theoretical and EXFOR definition of the 'm' and 'n' isomer.

Equivalence with exclusive channel cross-sections

There are a few classes of reactions where an exclusive cross-section is exactly equal to a residual production cross-section since there are other pathways to the residual nucleus. They are:

- (n,γ);
- (n,n'), (n,2n), (n,3n),.....
- (n,p), (n,2p), (n,3p),.....

Hence, if EXFOR contains reactions like ^{56}Fe(n,2n) and ^{56}Fe(n,x)^{55}Fe, it should be understood that both sets belong to the same total set which could be compared with each other and with nuclear model codes and data libraries. In these cases, the measurement method (measuring activation or the emitted particles) is important but considered as secondary metadata.

4.3. Angular distributions (single-differential)

Elastic scattering angular distribution

Definition of one data set

One data set for elastic angular distribution is defined by:

- the projectile;
- the target nucleus;
- one incident energy with its uncertainty;
- a set of outgoing angles;

- single-differential cross-sections with their uncertainties;
- all other metadata.

Relation to EXFOR

Similar complications as for discrete level cross-sections exist: Multiple incident energies, discrete level energies and emission angles may be present in an arbitrary sorted manner. There are subentries with one elastic angular distribution, i.e. one incident energy and a set of outgoing angles. It may, however, also occur that various incident energies are present in one subentry. The data in EXFOR may then be grouped by incident energy or by emission angle —all variations exist.

Discrete level scattering angular distributions

Definition of one data set

One data set for inelastic or discrete level angular distribution is defined by:

- the projectile;
- the target nucleus;
- the outgoing particle and level e.g.
  - (n,n_1)
  - (n,p_0);
- one incident energy with its uncertainty;
- a set of outgoing angles;
- single-differential cross-sections with their uncertainties;
- all other metadata.

Relation to EXFOR

Here, there might be an extra complication that the discrete level excitation energy is another running variable in the output block, in addition to the issues for elastic scattering.

4.4. Prompt, delayed and total fission neutron multiplicities

Definition of one data set

One data set for prompt, delayed and total fission neutron multiplicities is defined by:

- the projectile (e.g. neutron, proton, spontaneous fission);
- the target nucleus;
- the reaction type which can either be:
  - total (average number of neutrons originating from fission are counted);
  - prompt (average number of prompt neutrons from fission are counted);
  - delayed (average number of delayed neutrons from fission are counted).
- one or more incident energies with their uncertainties for particle-induced fission, or an identifier indicating spontaneous fission;
- multiplicities with their partial or total uncertainties;
- all other metadata.

Relation to EXFOR

This is probably among the easiest categories, in that often (but certainly not always) there is a one-to-one correspondence to a MEDUSAL data set and what is stored in one EXFOR subentry.

Relation to ENDF-6 format

These data would be identified in ENDF-6 format as:

- total: MF=1, MT=452;
- prompt: MF= 1, MT=456;
- delayed: MF=1, MT=455.

The evaluated covariances would be stored in MF=31 and the associated MT number.

4.5. Prompt and delayed fission neutron spectra

Definition of one data set

One data set for delayed, and prompt fission neutron spectra is defined by:

- the projectile (e.g. neutron, proton, spontaneous fission);
- the target nucleus;
- the reaction type which can either be:
  - prompt (only energy distributions of prompt neutrons are provided);
  - delayed (only energy distributions of delayed neutrons are provided);
  - total (energy distributions of both prompt and delayed neutrons are provided).
- one or more incident energies with their uncertainties for particle-induced fission, or an identifier indicating spontaneous fission;
- more than one outgoing neutron energies with their uncertainties;
- spectra with their partial or total uncertainties;
- all other metadata;
- parameters for Maxwellian and Watt distributions, if available.

In some cases, spectra are summarised in EXFOR and the literature by specifying the mean energy derived from the measured data. These mean energies often stand in lieu of the measured spectra. In the same vein, measured spectra can be summarised by fitting them to Maxwellian or Watt functions, or other functional forms representing fission neutron spectra. The parameters of these functional forms are then in some cases reported instead of experimental data. However, these functional forms are poor approximations of experimental data, and it would be preferred to actually list the experimental data as obtained from the measurement, if at all possible.

Relation to EXFOR

In some cases, there is a one-to-one correspondence to a MEDUSAL data set and what is stored in one EXFOR subentry. Fission neutron spectra have sometimes been categorised as such in EXFOR but also as single-differential emission spectra which has to be accounted for when converting data from EXFOR to MEDUSAL.

Relation to ENDF-6 format

These data would be identified in ENDF-6 format as:

- prompt: MF= 5, MT=18;
- delayed: MF=5, MT=455.

The evaluated covariances would be stored in MF=35 and the associated MT number.

4.6. Fission yields

There are three categories of yields for fragments and products generated during each process, namely primary (pre-neutron emission), secondary (post-neutron emission, but before beta decay), and cumulative (after beta decay). In the evaluated nuclear data library, the secondary and cumulative yields per fission are stored in MF8, MT454 and MT459, respectively, in the manner of `Y(Z, A, M)`, where `A` is mass number, `Z` is charge number, and `M` is the isomeric state. The sum of the fission yields, for each stage, should be normalised to 200%[3]. Generally, fission yields are measured for specific `Z`, `A`, or `(Z, A, (M))`. `Y(A)` is called mass yield and `Y(Z)` is called charge yield. The fission yield data could be given in units of particle or product per fission, or percent per fission (percent per 100 fissions). The most probable charge `ZP` is the most probable charge number in a given mass chain, and the most probable mass `AP` is the mean mass number of the fission fragments/products in light/heavy fragments or in the specific element. The other two quantities that are derived from the fission yields are fractional yields, the isomeric yield ratio. The fractional yields are given as a ratio of fission product yield to total chain yield `Y(A)`. The isomeric (yield) ratio (IR) is the relative population of the fission yields between the ground state and a long-lived excited state.The R-value is a ratio of two measurements one of which is well known standard reaction.

Primary fission fragment yield

The primary (pre-neutron) fission fragment is the fission fragment yield before prompt neutron emission. Note that the word "provisional", which means the pseudo mass before neutron emission determined from the 2E-method measurements, is also used.

Definition of one data set

One data set for primary fission fragment yield is defined by:

- the projectile;
- the target nucleus;
- the produced nucleus or particle charge `Z`, mass `A`, or `(Z,A)`;
- one or more incident energies with their uncertainties;
- yields with their uncertainties;
- all other metadata.

Relation to EXFOR

The fission yield is defined by `Y(Z,A)` and can be considered as `Z` = None if the data is `Y(A)` and `A` = None if the data is `Y(Z)`. Since ternary and quarterly fission can occur just after scission, ternary and quarterly fission yields are defined as primary fission fragment yields specified with the produced particle,which are usually very light particles such as tritium and helium isotopes. No IR measurements for primary fission fragments have been reported so far.

Independent fission product yield

Independent, or secondary, fission product yields are the yields post-neutron emission, but before beta decay, hence before delayed-neutron emission. The mass (number) yield of independent fission products was part of the chain yield but needs to be differentiated from the (total) chain yield defined for the cumulative yield as explained below.

[3]Except for the non-binary fission cases such as the ternary fission.

Definition of one data set

One data set for independent fission product yield is defined by:

- the projectile;
- the target nucleus;
- the produced nucleus or particle charge `Z`, mass `A`, or `(Z,A)`;
- the produced nucleus with the ground/isomeric state;
- one or more incident energies with their uncertainties;
- yields with their uncertainties;
- all other metadata.

Relation to EXFOR

Decay data used in the measurement are particularly important for evaluations, so the yields are frequently given with the decay data. In the case of R-values, two sets of mass/elements need to be defined in the data table and the data is given, and the data is generally not comparable to the direct measurements.

Cumulative fission product yield

The cumulative fission product yield is the yield after beta decay which leads the delayed-neutron emissions. Historically, the sum cumulative yield(s) of the last stable or long-lived isotope in the same mass chain is called the (total) chain yield.

Definition of one data set

One data set for cumulative fission product yield is defined by:

- the projectile;
- the target nucleus;
- the produced nucleus or particle charge `Z`, mass `A`, or `(Z,A)`;
- the produced nucleus with the ground/isomeric state;
- one or more incident energies with their uncertainties;
- yields with their uncertainties;
- all other metadata.

Relation to EXFOR

Same as in Section 4.6.2.

5. Data categories

In this section are requirements on the type of information that the database should be able to hold. These are based on the high level requirements mentioned previously, the type of information currently stored in EXFOR, and the information desired by the users detailed above. These are wide-ranging, and no individual database entry would have all of this information.

5.1. General containers

While the majority of the information specified so far in this document represents metadata that is specialised to the category (e.g. sample material), there are some types of information that show up all over the file. This includes text values, numerical values, links and references. For these, general data types should be defined that can act as a basic unit and can be used anywhere in the schema. The following general data types should be supported in the database.

Text Values

Text values can be single text strings, or an array of strings. The requirements of the text and text array data types are:

1. Text data types requirements
   - 1.1 Hold the string or array of strings.
   - 1.2 Identify the source of the information.
   - 1.3 Support description of where the information came from.

Numerical Values

Numerical values can be constants or 1-D arrays. The requirements of the numerical data types are:

2. Text data types requirements
   - 2.1 Hold the single number or array of numbers.
   - 2.2 Identify the source of the information.
   - 2.3 Support description of where the information came from.

Isotope

Isotopes appear frequently in the entries, so a general isotope data type should be created. The requirements of the isotope data type are:

3. Isotope data type requirements
   - 3.1 Hold the identifier of the isotope.
     - 3.1.1 Support natural abundance elements.
     - 3.1.2 Support specifying an excited level.
     - 3.1.3 Support particles such as neutrons and gammas.
   - 3.2 Identify the source of the information.
   - 3.3 Support description of where the information came from.

Link

In many cases the file will refer to information outside of the entry, such as DOIs for papers or references to data in other entries. For these, a `link` data type should be created. The information in this data type should allow the user to find the extra information easily. The requirements of the `link` data type are:

4. `link` requirements
   - 4.1 Identify the type of link.
     - 4.1.1 Support the types DOI, hyperlink, entry, evaluation, Nuclear Science Reference (NSR).
   - 4.2 Store the link.
   - 4.2 Identify the source of the information.
   - 4.3 Support description of where the information came from.

Measured Quantities

Measured quantities have both a unit and an uncertainty. A general data type should be created to hold measured quantities and the required metadata. The information in these data types should allow the user to propagate uncertainty and plot the values.

5. Measured quantity data type requirements
   - 5.1 Quantify the value(s) and uncertainty value(s).
   - 5.2 Identify the unit of the value(s) and uncertainty.
   - 5.3 Identify the type of value and uncertainty.
     - 5.3.1 Support the types mean, maximum, minimum, range and unresolved values.
     - 5.3.2 Support relative and absolute uncertainties.
     - 5.3.3 Support total statistical and total systematic uncertainties.
   - 5.4 Identify the source of the information.
   - 5.5 Support description of where the information came from.

Reference

References to papers, conference reports, etc. are used throughout the file. The information in the reference data type should allow the user to locate and cite the reference. The requirements of the reference data type are:

6. Reference requirements
   - 6.1 Identify the type of reference.
     - 6.1.1 Support the types journal, conference proceeding, institutional report, thesis, personal communication.
     - 6.1.2 If a journal or conference proceeding, identify the journal/conference and the year.
     - 6.1.3 If a thesis, identify the institution, department and year.
     - 6.1.4 If an EXFOR or NSR entry, identify the identification number in the database.
   - 6.2 Identify the title and authors of the reference.
   - 6.3 Identify a link to the reference, if available.
   - 6.4 Support free text describing the use of the reference.
   - 6.5 Identify the source of the information.
   - 6.6 Support description of where the information came from.

Correction

Corrections can be applied during data analysis for any of the data categories described above. All of the corrections can be described using a general correction data type. The requirements of the correction data type are:

7. Correction requirements
   - 7.1 Identify whether or not the correction was applied.
     - 7.1.1 Support fully corrected, partially corrected, not corrected, not corrected because the effect was negligible, and unknown.
     - 7.1.2 Support free text to describe whether or not the correction was applied.
   - 7.2 Identify the method(s) used for the correction.
     - 7.2.1 Support the methods fit, subtracted, measured, modeled, black-resonance filter, unknown.
   - 7.3 Identify the components of the experimental setup were included in the correction.
   - 7.4 Identify the code used for the correction.
   - 7.5 Identify whether the correction is energy-dependent, and if so, dependent on the incident or outgoing energy.
   - 7.6 Identify any reference data used in the correction.
   - 7.7 Quantify the value(s) and uncertainty of the correction.
   - 7.8 Support free text to describe the correction.
   - 7.9 Identify the source of the information.
   - 7.10 Support description of where the information came from.

Commercial Product

In many experiments, the detector and related electronics are commercial products that have serial numbers. In fully documented experiments, the manufacturer and serial number of commercial products are provided, and should be compiled in the database. For this reason, a general data type should be created that represents a commercial product. The requirements of the commercial product data type are:

8. Commercial Product requirements
   - 8.1 Hold the manufacturer of the product.
   - 8.2 Hold the serial number of the product.
   - 8.3 Identify the use of the product.
   - 8.4 Support description of the product in free text.
   - 8.5 Identify the source of the information.
   - 8.6 Support description of where the information came from.

5.2. Observable

The most general category is the observable category, which should hold all of the basic information about observable and the history of the EXFOR compilation. The information in this container should allow the user to identify what other data sets or models to compare this data to. The requirements of the `Observable` container are:

9. `Observable` requirements
   - 9.1 Describe the observable of the experiment in a unique way that is understood and usable for all use cases discussed.
   - 9.2 Identify the MFMT number the observable corresponds to, if applicable.
   - 9.3 Identify whether the experiment results are absolute or relative.
   - 9.4 Identify the reference frame of the results.
   - 9.5 Identify whether the incident particles were polarised.
   - 9.6 Identify the basic method(s) used in the experiment.
     - 9.6.1 Support all EXFOR analysis codes and any included free text.
     - 9.6.2 Support all EXFOR method codes and any included free text.
   - 9.7 Identify the dependencies of the incident particles and the measured particles.
   - 9.8 Hold all of the parts of the EXFOR reaction string.
   - 9.9 Hold all of the numerical values in the EXFOR entry needed to plot the results.

5.3. Bibliography

The bibliography container holds the information about the references, authors, institute and original EXFOR entry. The information in this category should allow for the user to identify where to look for additional information or insight about the experiment and results. The requirements of the `Bibliography` container are:

10. `Bibliography` requirements
    - 10.1 Identify the original EXFOR accession number for the entry and the entry number in the MEDUSAL database.
    - 10.2 Identify the main reference, as well as any relevant references.
    - 10.3 Identify the institute(s) where the experiment took place.
      - 10.3.1 Support all EXFOR institute codes and any included free text.
    - 10.4 Hold the general information about the EXFOR entry, such as the history of the compilation.
      - 10.4.1 Support all EXFOR status codes and any included free text.
      - 10.4.2 Support all EXFOR comments codes and any included free text.
      - 10.4.3 Support all EXFOR critique codes and any included free text.
    - 10.5 Identify whether the data in the entry has been altered compared to the data in the main reference.

5.4. Incident particles

The `incident` container holds the information about the incident particles. The information in this container should allow the user to fully characterise the incident particles. The requirements of the `Incident` container are:

11. `Incident` requirements
    - 11.1 Identify the name and type of the facility, the source, and the reaction used to produce the incident particles.
      - 11.1.1 Support all EXFOR facility codes and any included free text.
      - 11.1.2 Support all EXFOR inc-source codes and any included free text.
    - 11.2 Identify the flux, fluence, or current determination method, values, and any relevant corrections.
      - 11.2.1 Hold the flux, fluence, or current values and uncertainties.
      - 11.2.2 Identify and quantify any corrections made for fluctuations in the flux, fluence, or current between runs within the experiment.
      - 11.2.3 Identify and quantify any corrections made for the uniformity of the incident beam across the face of the sample.
      - 11.2.4 Identify whether the beam or accelerator stability were monitored during the experiment.
    - 11.3 Identify the incident particle spectrum determination method, values and any relevant corrections.
      - 11.3.1 Hold the energy spectrum values and uncertainties.
      - 11.3.2 Identify and quantify any corrections made for secondary energy groups, and describe the secondary energy groups.
      - 11.3.3 Identify the method used to determine the incident particle spectrum.
        - 11.3.3.1 For TOF-based methods, identify and quantify (where applicable) the flight path length, macro pulse width, micro pulse width, repetition rate, zero time offset, beam energy, beam power, beam dimensions, and any filters, collimators or moderators used.
        - 11.3.3.2 For quasi-monoenergetic particle sources, identify and quantify (where appli--cable) the particle detected, the target thickness, the target number density, the target stability, the gas cell pressure and temperature, corrections on the gas cell and corrections for anisotropy.
    - 11.4 Identify any filters, collimators and moderators used in the experiment.

5.5. Detectors

The `detectors` container holds information about all detector systems used in the experiment. This includes any detector systems used for the observable, monitor reactions, decay measurements, and flux/fluence/current measurements, etc.

The requirements of the `Detectors` container are:

12. `Detectors` requirements
    12.1 Hold an unlimited number of detectors, each in a `detector` container.

The `Detector` container describes a single detector system in the experiment. A detector system can be a single detector or a collection of detectors that are used together to measure a single observable. The information in this container should allow the user to fully characterise the detectors used and their uncertainties. The requirements of the `Detector` container are:

13. `Detector` requirements
    13.1 Identify the use of the detector system in the experiment and the particle detected.
    13.2 Identify the number of individual detectors or segments in the detector system.
    13.3 Identify the type of detector.
        13.3.1 Support all EXFOR detector codes and any included free text.
        13.3.2 If a gas detector, identify the type of gas and quantify the pressure of the gas.
        13.3.3 If a solid state detector, identify the reaction used to detect the particle, the material of the detector, and quantify any correction for stopping power.
        13.3.4 If a fission detector, identify the fissionable isotope used for detection and quantify the correction for pulse height extrapolation.
    13.4 Identify the detector or segment dimensions.
    13.5 Identify the manufacturer of the detector and Photomultiplier Tube (PMT), or whether they were created in-house. Identify the Data Acquisition (DAQ) systems used and their manufacturers.
    13.6 Identify the method used for efficiency calibration and energy calibration, and any calibration sources used for either. Identify any shielding used, and the solid angle of the measurement. Quantify the efficiency and energy resolution values and uncertainties.
    13.7 Identify and quantify the efficiency corrections for pulse pile-up, the discriminator, electronic noise, forward boost, angular distortion, delayed gammas, false fission, the French effect, Hs content, manganese capture, false capture, sample radioactivity, spectrum unfolding, sum peaks, gamma angular distributions and counting geometry.
    13.8 Identify and quantify corrections for deadtime and detector live time.
    13.9 Identify whether the measurement was done in coincidence and the time gate used, if so.

5.6. Samples

The `samples` container holds information about all samples measured in the experiment. This includes any sample used as a monitor reaction, to determine the flux, etc. This does not include the targets used in the production of the incident particles.

The requirements of the `Samples` container are:

14. `Samples` requirements
    - 14.1 Hold an unlimited number of samples, each in a `sample` container.
    - 14.2 Identify the configuration of the samples if they are measured simultaneously.

The requirements of the `Sample` container are:

15. `Sample` requirements
    - 15.1 Identify the use of the sample in the experiment.
    - 15.2 Identify the sample material, geometry and phase.
    - 15.3 Identify the isotopic composition of the sample and quantify any corrections for impurities.
    - 15.4 If a unique incident energy is seen by the sample, quantify it.
    - 15.5 Identify the position of the sample if it is measured simultaneously with other samples.
    - 15.6 Identify the production method of the sample.
    - 15.7 Quantify the dimensions of the sample.
    - 15.8 Identify any containers or backings on the sample.
    - 15.9 Quantify the particulate radius of the sample.
    - 15.10 Quantify the thermodynamic and effective temperatures of the sample.
    - 15.11 Quantify all relevant times for radioactive samples, including the irradiation time, cooling (waiting) time and the counting time.
    - 15.12 Identify and quantify corrections made for uniformity, roughness, displacement, beam overlap, gamma attenuation, resonance self-shielding, multiple-sample effects and radioactivity.

5.7. Response function

For some experiments, the response function of a detector system or of the entire experiment maybe be necessary to fully understand and utilise the data set. For example, for resolved and unresolved resonance region measurements, the resolution function of the measurement is needed in order to properly use the experimental transmission and capture yield data. The information in this container should allow the user to fully characterise the resolution function for use in resonance region fitting programs. In other cases, the response function of the detector system can be defined separately. The requirements of the `Response` container are:

16. `Response` requirements
    - 16.1 Identify the type of resolution function.
      - 16.1.1 Support resonance region resolution functions and detector system response functions.
    - 16.2 Identify the format of the resolution function.
      - 16.2.1 Support equations with input values, tabulated values and program input files.
      - 16.2.2 For equations, quantify the parameter values and identify the equation.
      - 16.2.2 For program input files, identify the program used and link to the file.
    - 16.3 Support free-text description of the resolution function in addition to the values.
    - 16.4 Reduce redundancy errors by linking to other values in the entry where possible.

5.8. Data sources

Other nuclear data is used in many experiments. This can take the form of decay data, reaction cross-sections, PFNS, etc. All of the reference data used should be documented in the `datasources` container, with each separate source of reference data in a `datasource` container. The requirements of the `Datasources` container are:

17. `Datasources` requirements
    17.1 Support an unlimited number of `datasource` containers.

The requirements of the `Datasource` container are:

18. `Datasource` requirements
    18.1 Identify the particle detected for the comparison to the reference data.
    18.2 Identify the reference for the data and link to the values if possible. If not, quantify the values used in the experiment.
    18.3 Identify the type of data.
        18.3.1 Support the types: reaction cross-section, decay data, atomic mass, PFNS, thermal capture gamma intensity.
        18.3.2 For reaction cross-section, identify the reaction.
        18.3.3 For decay data, identify the type of decay, the parent and daughter isotopes.
        18.3.4 For PFNS, identify the isotope, whether the spectrum is described by a Watt or a Maxwell-Boltzmann function, and quantify the relevant parameter.
        18.3.5 For atomic mass, identify the isotope and quantify the mass used.
        18.3.6 For thermal capture gamma intensity, identify the target isotope and the energy of the gamma used, and quantify the intensity value used.

5.9. Correlation to other data sets

There are often correlations between different entries that were measured at the same facility, with the same sample, and/or using the same reference data, etc. The `correlations` container should hold the individual sources of correlation, with each source in a separate `correlation` container. There may be multiple `correlation` containers describing the correlation between two entries, if there are multiple sources of correlation. The information in this container should allow a user to correctly create a correlation matrix between data sets. The requirements of the `Correlations` container are:

19. `Correlations` requirements
    19.1 Support an unlimited number of `correlation` containers.

Each source of correlation will be described with a `correlation` container. The information in this container should allow a user to correctly include this source of correlation in a covariance matrix between the two data sets. The requirements of the `Correlation` container are:

20. `Correlation` requirements
    20.1 Identify the entry.
    20.2 Identify the factor(s) that correlate two data sets.
        20.2.1 Support individual keywords and entire containers.
    20.3 Identify the shape of the correlation (in incident energy) and quantify the correlation values.

6. IAEA nuclear data section EXFOR tools and databases

Due to the long history and importance of the EXFOR database, it has prompted numerous tools and derived databases. This section focusses on a subset of the work at the IAEA that was focussed on by SG50.

6.1. Development of python EXFOR parsers

The IAEA Nuclear Data Section (NDS) is responsible for maintaining the EXFOR database (Otuka et al., 2014) and providing user-friendly access to it. New use cases, e.g. involving machine learning, necessitate well-documented APIs and data formats that can be efficiently handled by the user. Until recently, the retrieval of a large number of datasets still required manual intervention, which is an obstacle to working efficiently with the experimental data. Furthermore, the retrieval from the EXFOR database is only one of several required steps (commonly referred to as the nuclear data pipeline) to make nuclear data usable in nuclear applications, such as reactor simulations. Modern APIs and data formats are essential for enabling an efficient flow of nuclear data through these steps.

Therefore, to facilitate access to the data in the EXFOR database, Shin Okumura and Georg Schnabel from the IAEA-NDS have recently (in 2021-2024, see Okumura et al., 2024) developed two Python-based EXFOR parsing programs for translating EXFOR-formatted data into the JSON format, which are:

1. `exfor-parserpy` (https://github.com/IAEA-NDS/exfor-parserpy): Conceived as a proto--type within the framework of WPEC SG50 to explore the suitability of various programming concepts for the creation of an EXFOR parser.
2. `exforparser` (https://github.com/IAEA-NDS/exforparser): Developed independently within a currently ongoing IT modernisation initiative at the IAEA-NDS, serving as a building block for the development of new web services and APIs at the IAEA-NDS.

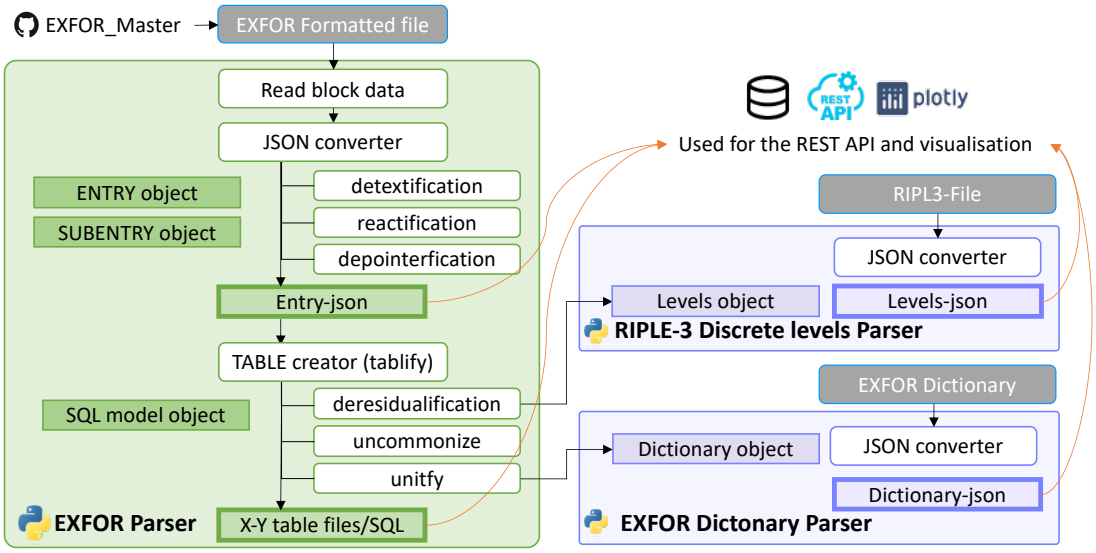
`exforparser` was developed by Shin Okumura and `exfor-parserpy` mainly by Georg Schnabel, with contributions from Shin Okumura and Paul Reiter.

This chapter summarises the development of these programs, their functionalities, the processing of EXFOR-formatted files, and the resulting improvements in data management and access through web interfaces and REST APIs. A flow chart visualising the processing sequence starting from EXFOR master files shown in Figure 6.1, the steps and components for which will be explained in the following sections.

EXFOR Master Files

The EXFOR parsers are programs to parse EXFOR Master Files. These files contain experimental data and associated rich meta data about the details of the experiments. The entire collection of EXFOR master files represent the core of the EXFOR database. In the past, this collection was not available for download but only subsets could be manually retrieved by a search query launched over a graphical user interface (GUI), provided as a web service. More recently, it was recognised that this restriction is not aligned with FAIR principles. FAIR is an acronym that stands for Findable, Accessible, Interoperable and Reproducible. For instance, some users

Figure 6.1: Flow chart visualising the processing of information from the EXFOR database enabled by the developed EXFOR parsing programs into more user-friendly formats and services.



may want to design their own data extraction pipelines and use a local copy of the entire, original EXFOR database on their computer. Therefore, to address such use cases and make EXFOR more FAIR, the entire collection of EXFOR master files has been made available as a Git repository at https://github.com/IAEA-NDS/exfor_master/.

EXFOR format

The purpose of this section is to provide a brief explanation of how the experimental data are organised and stored according to the EXFOR format, to better understand the description of the EXFOR parser functionalities summarised afterwards. The full details of the of the EXFOR format can be found in the EXFOR formats manual (Schwerer, 2015).

The EXFOR format defines a hierarchical structure, with the top-level organisation established by `ENTRY` sections associated with an `ENTRY` number. These blocks are further divided into `SUBENTRY` sections identified by their `SUBENTRY` number. Below that, three sections can be present, which are a `BIB` block for bibliographic information, a `COMMON` block for common numeric values associated with all measurement points and a `DATA` block for the measured data. Within the `BIB` section, each key is associated with a string, whereas the `COMMON` and `DATA` block contain only numbers (with the exception of column headers containing the specification of the physical quantity and physical unit used). The EXFOR format also specifies how this logical organisation is represented within a file. The data and meta data are stored as ASCII text with each line spanning at most 66 characters. All keys, such as keys in the bibliographic section or column headers in the `DATA` block and all numeric data, are stored in fields with 11 character slots. If the data associated with a key is of type text (only the case in the `BIB` section), the text field is associated with 55 character slots. In addition to issues in the logical organisation that complicate efficient access to data, a significant barrier to accessing the data is the way they are physically stored in a file.

The utility of the developed EXFOR parsers lies in the fact that they map the original data (given in a somewhat cumbersome physical representation in a file) into a data structure useable in Python. Thereby Python users can directly work at the the level of the logical organisation,

separated from the file representation. By extension, the conversion to other data container formats, such as JSON, becomes much easier.

Summary of the EXFOR parsers functionalities

In the following paragraphs, the functionalities implemented in the EXFOR parsers have been summarised. Because the two parsers, `exforparser` and `exfor-parserpy`, share many common design aspects, a general description has been provided that is valid for both.

The core functionality of the parsers is to read the data from an EXFOR-formatted file and store it in a Python dictionary. A Python dictionary is a data structure that allows for the storage of arbitrary objects under `keys` (strings), enabling the creation of flexible hierarchical data structures. Therefore, the logical hierarchical data structure defined by the EXFOR format can be seamlessly and comprehensively translated into a (nested) Python dictionary.

Even though this basic conversion already greatly facilitates the access to raw EXFOR data, the resulting data structure can be further post-processed and transformed for an improved user-experience. One example of such a transformation is to unify all physical units. In the following, a detailed description on post-processing steps implemented in the parser programs is provided.

- Decomposition of reactions strings into interpretable components: The string stored under the key `REACTION` in the bibliographic section fully specifies the reaction system investigated (projectile and target) and the type of observable being measured (e.g. cross-section or angular distribution). Splitting the string up into several components enables a user to directly access essential information, such as the projectile type, under its own key without the need to parse a string.
- Separation of EXFOR codes and free text: Text fields in the bibliographic section usually contain an `EXFOR code` given in brackets and free text afterwards. An example is the `INSTITUTE` field that would contain as EXFOR code an abbreviation of the institute name where the measurement has been performed. Available EXFOR codes are registered in so-called EXFOR dictionaries. The free text, on the other hand, is not governed by any restrictions and is often (but not always) used as a human-readable version of the EXFOR code preceding it. Splitting the EXFOR codes and free-form text facilitates the creation of programs interpreting EXFOR data.
- Integration of COMMON blocks into DATA tables: `COMMON` blocks contain numerical values that are associated with all measurement points recorded in the table stored in the `DATA` section, such as a fixed angle for a measurement. While this EXFOR feature is reasonable to avoid unnecessary duplication, it leads to a more involved logic of programs that need to interpret the data, e.g. by merging the numbers from the `COMMON` block into the `DATA` table before plotting. Therefore, this post-processing step is also implemented in the parser programs.
- Splitting datasets referring to several physical observables: The EXFOR format supports a feature referred to as pointers. These pointers allow for the storage of experimental data associated with different reaction systems or observable types. This is accomplished by putting a single digit or letter (the pointer name) into the last character slot of a key field. While there are reasons from the experimental point of view to store different measurement campaigns in a single dataset, e.g. because they were measured at the same facility, it makes the automatic interpretation and disentanglement of such an EXFOR dataset more difficult. The parser programs allow for distribution of the information on

the different measurements in a single dataset into several datasets. In this way, pointers can be eliminated and each dataset corresponds to exactly a single reaction system and observable type.
- Unification of measurement units: All measurement units are explicitly specified in table headers in the EXFOR format. Having different datasets using different units requires user programs to implement extra logic to unify all units. As the unification of all units is a very common task that needs to be performed before many interactions with EXFOR data, the parsers also support the unification of the unit as a post-processing step. Users or down-stream programs can then rely on the assumption that all units are standardised.

In summary, the core functionality of the parsing programs is to read the experimental data from EXFOR-formatted files and store them in a Python dictionary. Several post-processing steps, some optional, can be performed to make the data access more user-friendly.

Conversion to JSON and tabular formats

While Python dictionaries are convenient for Python programmers, they cannot be accessed and used in other programming languages. Therefore, the parsed EXFOR data needs to be stored in a format that is widely supported in the domain of information technology.

The JSON format is a long-established standard format for the transmission of data over the internet. Due to its widespread use, most popular programming languages provide facilities for accessing and manipulating JSON files. Therefore, in accordance with FAIR principles, the parsing programs also support the storage of the Python dictionaries as JSON files. It is noteworthy that this translation is quite trivial to implement with the `json` package available in Python. These JSON files with EXFOR datasets can be directly used by user programs or fed into a document-oriented database to enable versatile searching (Schnabel, 2020).

While JSON files are great for storage, many data science workflows rely on data in tabular form, e.g. `keras` in R (Chollet, Allaire, et al., 2017) and `scikit-learn` in Python (Pedregosa et al., 2011). The extraction of tables of measured data is a useful feature for many users and is supported by both parser programs. The resulting tables are provided in a standardised format with columns `(x, y, dx, dy)`, which is more useful for advanced analysis in various nuclear physics applications.

Augmenting EXFOR data with information from other sources

The EXFOR format itself does not define valid choices of EXFOR codes, such as abbreviations of institutes. Valid choices of EXFOR codes, abbreviations of physical quantities and units are recorded in EXFOR dictionaries. Each dictionary is associated with a specific category of information, e.g. institute names, and contains an ASCII list of all valid options.

Another source of information is the Reference Input Parameter Library (RIPL) (Capote et al., 2009) containing tables of masses, discrete levels and other essential information for nuclear model codes. The RIPL library can also be employed to match the energies given for inelastic scattering in the EXFOR database to the discrete level states stored in the RIPL library.

The augmentation of EXFOR data with information from both the EXFOR dictionaries and RIPL library is useful to users. Dictionaries can be used to expand abbreviations to the full name and the addition of discrete level numbers enables more efficient searching and filtering of experimental data. The `exforparser` package already leverages these additional sources of

information, whereas the `exfor-parserpy` package does not.

EXFOR dictionary in JSON

EXFOR Dictionaries contain the descriptions associated with the various EXFOR codes used in the EXFOR database. For instance, EXFOR codes for detectors specified under the keyword `DETECTOR` are (among others) `GELI`, `SCIN`, and `HPGE`, which indicate a Germanium detector (GELI), Scintillation detector (SCIN), and High-Purity Germanium detectors (HPGE), respectively.

To leverage this information, Shin Okumura from the IAEA-NDS also developed a program to convert an EXFOR Dictionary into the JSON format. By transforming dictionaries into JSON, definitions of EXFOR codes can be more easily accessed. For instance, accessing the definition of a Germanium detector can be simply done by `exfordict['DETECTOR']['GELI']` with `exfordict` being the Python dictionary containing the EXFOR dictionary information.

By using the EXFOR Dictionary in JSON, a user can quickly retrieve precise definitions and descriptions of the terms without sifting through extensive documentation. This streamlined access supports more efficient data analysis and experiment documentation, ensuring that users can focus on their research rather than on interpreting complex formats. The EXFOR Dictionary in JSON is maintained on the IAEA-NDS GitHub account, available at https://github.com/IAEA-NDS/exfor_dictionary/.

REST APIs and web interfaces

To facilitate seamless data access and visualisation, Shin Okumura from the IAEA-NDS developed REST APIs and user-centric web interfaces. These advanced tools enable responsive interactions between client applications and the database, enhancing overall user experience and data utilisation. Key features of our API and web interface include:

- EXFOR entry preview web user interface (https://nds.iaea.org/dataexplorer/exfor/);
- interactive plots for data visualisation based on nuclear physics (https://nds.iaea.org/dataexplorer/);
- downloadable datasets in text format;
- REST API endpoints for EXFOR datasets, EXFOR dictionary, and RIPL-3 discrete levels (https://nds.iaea.org/dataexplorer/api/).

These developments are recent additions to the many IAEA web services already provided by the IAEA-NDS and are recently developed alternatives to some of the existing web services, such as the EXFOR web interface (V. V. Zerkin & Pritychenko, 2018).

6.2. EXFOR dissemination systems

The EXFOR database is maintained by the NRDC, which is co-ordinated by the IAEA. The source of EXFOR is the database of EXFOR Master Files, which is updated and maintained by the IAEA-NDS. In the past the EXFOR Master Files were not available outside of the NRDC community, but a change was made at the 2023 NRDC meeting (Otuka & Brown, 2023):

> "The NRDC supports releasing the EXFOR Master Files, Dictionaries and their documentation as Open Data, with Document Object Identifiers (DOI) and an acceptable open data license (CC-BY-4.0 or similar). Each released Master File would then require its own DOI and internet landing page. This distribution should be

> made retroactively ca. 2015 onward and should not include NRDC working materials such as preliminary and final trans tapes, and backup files."

The files and dictionaries are now available as text files in various repositories on the IAEA-NDS github.

The NDS offers web retrieval services for the EXFOR database, which allows users to search through the database and access individual datasets. To support this, the EXFOR files have been parsed into a relational database (V. V. Zerkin & Pritychenko, 2018) by Viktor Zerkin from the IAEA-NDS. Recently, the relational database has been extended to include the experimental data points, accessible using SQL commands, in the X4Pro database (V. Zerkin, 2023). The X4Pro database allows for renormalisation based on monitor values and decay data values and applying corrections to the data values. Another conclusion of the 2023 NRDC meeting was to distribute EXFOR data in the form of the X4Pro database (Otuka & Brown, 2023):

> "NRDC supports proposal of Zerkin to distribute X4Pro database with X5 as a product of NRDC recommended for users' community. Implementing must include sharing NDS source code and documentation producing X4Pro within NRDC."

The X4Pro database is currently available from the IAEA-NDS website (https://www-nds.iaea.org/cdroms/#X4Pro) as a mini-distribution (with an incomplete EXFOR database) and as a full distribution. Both distributions include the EXFOR database in an SQLite database and demo programs in Python, C and Fortran.

X5json, "comprehensive EXFOR in JSON", is currently available through the EXFOR Web retrieval system (as one of the output formats), through Web-API, and as a test version of the full EXFOR database on GitHub (https://github.com/vzerkin/EXFOR-X5json). X5json presents both original and computational formats of the EXFOR data, and includes information from the EXFOR Dictionaries, the RIPL database (Capote et al., 2009), and monitor and decay data for renormalisation (see V. Zerkin, 2024, p. 6 and V. V. Zerkin, 2023, p. 40).

Web-API is implemented via GET requests to the IAEA-NDS Web server (https://nds.iaea.org/exfor/x4guide/API/), allowing a user to search and download data from EXFOR and various evaluated libraries. Some calculations can be performed on the server, including for data renormalisation. The data are presented in the original formats (EXFOR, C4, C5) and also in JSON, XML, CSV.

7. Use case for employing SG-50 formatted data for machine learning

7.1. Background

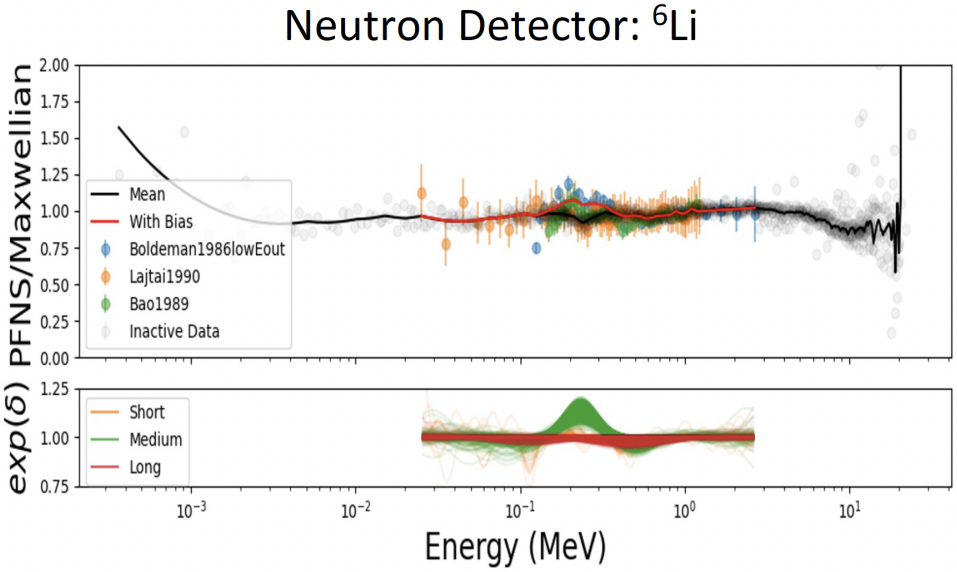
The AIACHNE (AI/ML Informed cAlifornium CHi Nuclear data Experiment) project aims at designing an experiment for the ^{252}Cf Prompt Fission Neutron Spectrum (PFNS) that explores systematic biases in an experimental database retrieved from the EXFOR databases. To that end, ML methods were applied to pinpoint measurement features likely related to bias (Walton et al., 2024). From that information, a feature was selected that should be explored by the AIACHNE experiment.

Measurement features are metadata encapsulating all pertinent information about the physical measurement and analysis techniques. Examples are, for instance, what neutron and fission detectors were used for the physical measurement, and what background reduction techniques were employed for analysis techniques. Such metadata were retrieved both from EXFOR entries as well as the literature of data sets described in detail in Reference (Neudecker et al., 2024).

The prerequisite for applying machine learning techniques is casting the metadata into a format that can be parsed by the algorithm. This step might seem trivial but requires finding a unique language where metadata that carry the same physics meaning across several experiments must have the same identifier. One example is, for instance, the neutron detector. As seen in Figure 7.1, the machine learning code identified the use of ^{6}Li detectors as being related to bias in some datasets of the AIACHNE ^{252}Cf PFNS experimental database. In fact, there are several experiments that used neutron detectors containing ^{6}Li in the database, for instance References (Bao et al., 1989; Boldeman et al., 1986; Lajtai et al., 1989). The EXFOR format has unique keywords describing detectors such as “SCIN” or “GLASD”. One may think that these keywords are already sufficient descriptors for ML to uniquely find an issue. However, “SCIN” (used for Bao et al., 1989; Boldeman et al., 1986) and “GLASD” (used for Lajtai et al., 1989) fail to inform the algorithm what the active material in the detector is. And, the key common issue leading to bias in ^{252}Cf related to neutron detectors is not whether it is a glass detector or a scintillator. Rather, the issue is that ^{6}Li was within both detector types and that even small mistakes in the detector response functions around approximately 200 keV are amplified by the ^{6}Li(n,α) resonance there leading to bias in data as highlighted in Fig. 7.1 and in (Walton et al., 2024). Hence, the features describing the neutron detector must call out the active material in the detector, rather than the existing EXFOR detector keyword, that the ML algorithm can find physically meaningful features related to bias.

Note: Figure 7.1, is an example of an AIACHNE ML output (Walton et al., 2024) that shows in the upper panel PFNS data, where the labeled data points in various colors except grey were measured with a neutron detector employing ^{6}Li as active material. The data of Boldeman (Boldeman et al., 1986) and Bao (Bao et al., 1989) show a clear bias that is identified as of medium range (across neutron energy) in the lower panel. The meaning of the entire plot is that the data of Boldeman et al. (1986) and Bao et al. (1989) , show a biased behaviour around approximately 200 keV that is related by the ML algorithm to the use of a ^{6}Li detector. While Lajtai et al.(1989) also used a ^{6}Li glass detector, they made considerable efforts to correctly resolve the detector response function, and therefore, do not show the bias.

Figure 7.1: An example of the AIACHNE ML output (Walton et al., 2024)



The AIACHNE team used a precursor of the WPEC (Working Party on International Nuclear Data Evaluation Co-operation) SG(Subgroup)-50 format to store the metadata for the ML analysis. Below, a summary is provided of the individual steps that were required to create the database as an example of what it means to create a curated database as envisioned by WPEC SG-50.

7.2. Creating the curated database

In a first step, experimental data for the ^{252}Cf PFNS were downloaded from the EXFOR database and compared to those data used in the previous ^{252}Cf PFNS neutron standards evaluation by W. Mannhart (Mannhart, 1987). Those data that were not yet in EXFOR were read from plots in journal articles. For the ML codes, the following data were needed:

- metadata features in a unique format;
- experimental mean values as a function of outgoing neutron energies;
- total covariances for the experimental data containing all pertinent uncertainties.

The experimental metadata features were extracted either from EXFOR entries or the literature. They were first all recorded in an excel spreadsheet. While this step might seem antiquated, it helps compare features across experiments. Comparing features across experiments is key to finding unique keywords that use the same identifier for the same physics effect, correction, analysis technique, type of equipment, etc., across all experiments of the database. Then, the metadata were cast into the precursor WPEC SG-50 JSON format using the code ARIADNE (Neudecker, Denise, 2018).

Experimental mean values and outgoing energies were either retrieved directly from the EXFOR database or read from the journal article. Total experimental covariances were estimated with the code ARIADNE as described in the following references (Neudecker et al., 2024; Neudecker, Denise, 2018). These covariance were estimated based on partial experimental uncertainties found in the journal articles or in EXFOR as described in (Neudecker et al., 2024). If no values were given for a particular uncertainty source that is expected to appear for a measurement, these

missing uncertainties were estimated based on templates of expected measurement uncertainties for PFNS (Neudecker, Devlin, Haight, et al., 2023).

7.3. Example from the curated database

Below, an example for one data set in the database is given for the experimental data of Chalupka et al. (Chalupka et al., 1990). This example would usually be stored in layer three of WPEC SG-50 as it contains total curated covariances, and comments by the evaluators.

There are two main parts to the JSON file: The part "attributes" holds all the metadata features. The second part is termed "data" and holds all numerical values for the experiment.

The metadata can be further sub-divided into the broad categories that describe the observable, the bibliography, the detectors (fission and neutron detectors), sample, shadow bar descriptors and corrections (including if corrections have been undertaken or which method has been applied). Contrary to the specification document, metadata for the "incident", "response functions", or "reference data" container are not given as these do not apply. The ^{252}Cf PFNS is for spontaneous fission without incident particles. The response function is considered within the analysis for multiple scattering and detector efficiency, and no reference data apply to the whole data set. However, important metadata are added as part of curation like "documentation", missing corrections or concerns by the team which WPEC SG-50 would like to have stored in layer three of the database. Also, it is important to record if corrections were undertaken and what methods were applied for curation for evaluations. If an important correction is missing, that might bias the data.

The "data" section contains the actual mean values (the shape was rescaled to Mannhart's evaluation Mannhart, 1987) with outgoing neutron energy, correlations and uncertainties. For the energies (incident and outgoing), the energy grid is provided along with units.


```
{
    "attributes": {
        "type": "DifferentialExperiment",
        "particle": "decay",
        "reaction": "PFNS",
        "nuclide": "Cf-252",
        "library": "ARIADNE",
        "documentation": "EXFOR No. = 22202.003+22202.004+22202.005. Journal
            article = Chalupka et al., Nuclear Science and Engineering, Vol.106,
            Issue.3, p.367 (1990).The data by Chalupka et al. are in EXFOR
            sub-entries 22202.003 (9--12.76 MeV), 22202.004 (13.98--27.10 MeV),
            22202.005 (9--15.52 MeV), 22202.006 (9--18.94 MeV), and 22202.007
            (9.99--20.02 MeV). The different EXFOR entries correspond to
            different energy ranges or bias settings (2.0, and 2.75 MeV). They
            can be combined considering correlations if not the same or
            overlapping energy ranges are used; otherwise, one would double count
            data as they stem from the same measurement. The journal publications
            were published after Mannhart's evaluation, but he considered these
            data along with those of Maerten et al. for inclusion.The main aim of
            this measurement was to counter-check the upturn of the PFNS above 20
            MeV seen by the Maerten et al. group which was contrary to the
```

```
        evaluation by Mannhart. They identify as main potential issue in that
        measurement above 20 MeV background from cosmic radiation. In order
        to reduce this specific background, Chalupka et al. performed their
        measurement in a mine in Bad Bleiberg, Austria which is 1000 m below
        a mountain known as a rich harvesting grounds for lead and tin. In
        essence, they had a 1000-m thick natural Pb shield to reduce
        random-coincidence background. One 25.4x5.08-cm scintillation
        detector of type NE213 was used for all runs to count neutrons. The
        efficiency of this detector was low (20%). The efficiency values were
        calculated by the MC technique using NEFF-4 up to 20 MeV. For 18-29
        MeV, they used the 1H cross-section data adjusted to NEFF results
        between 16-18 MeV, which agree well with today's JENDL-5.0 data. The
        light output curves were calibrated with 22Na, 60Co, and 208Tl
        Compton edge energies The TOF path was 279.6 cm long and had
        uncertainty of 2 mm. The neutron detector was located at 0 degree
        relative to the perpendicular plane of the 252Cf deposit. Fission
        fragments were detected in an  99.5% efficient fission chamber. The
        252Cf deposit was a layer of 15 mm diameter on platinum backing with
        15 mm diameter; it had an activity of 1.5643x10e5 f/s. The time
        resolution of the detection system was 1.0 ns FWHM. It could be that
        the detectors used were those used for the measurement of Boettger et
        al.Corrections were made for all known effects, and detailed partial
        uncertainties were provided for all uncertainties either explicitly or
        stating that they were 0.There were some minor concerns: One was that
        the carbon cross-section used in NEFF-4 agreed with JENDL-5.0 only to
        7% (total) and 9% (inelastic), however, the influence should be small
        as the main cross-section used if H. Also, it was not completely clear
        in how far the 252Cf PFNS spectrum was used for the detector response.
        The authors of this data set could, unfortunately, not clarify this
        anymore. As these are minor concerns, and many important corrections
        were performed and detailed uncertainties are provided, we accept the
        data until 25 MeV, which is the limit of our evaluation.",
    "date": "2024-05-22",
    "Author": ["A.Chalupka", "L.Malek", "S.Tagesen", "R.Boettger"],
    "Year": "1990",
    "Facility": "N/A",
    "Institute": ["2AUSIRK", "2GERPTB"],
    "Isotope": "Cf-252",
    "Quantity": "PFNS",
    "MeasurementType": "PFNS",
    "ConcernEXFOR": "N/A",
    "ConcernAIACHNE": ["Response function", "Multiple scattering"],
    "ConcernAIACHNE_text": "We are not sure how Maxwellian of T=1.42 was
        used to get the detector response. There could be a circular argument
        there. There could be scattering from the shadow cone leading to
        neutrons from water and iron. Gammas will have been taken care of by
        PSD cuts.",
    "BackgroundCorrected": "y",
    "AlphaBackground": "y",
```

```
"GammaBackground": "y",
"RandomCoincidenceBackground": "y",
"ShadowBarBackground": "y",
"WrapAroundBackground": "N/A",
"MultipleScatteringSampleBackingCorrected": "y",
"MultipleScatteringSurroundingCorrected": "y",
"AttenuationSampleBackingCorrected": "y",
"AttenuationSurroundingCorrected": "y",
"FissionDetectionEfficiencyCorrected": "y",
"NeutronDetectionEfficiencyCorrected": "y",
"NeutronDetectionResponseCorrected": "y",
"SampleDecayCorrected": "unknown",
"FissionFragmentAbsorptionInSampleCorrected": "y",
"SignalPulsePileupCorrected": "y",
"DeadtimeCorrected": "y",
"AngularDistributionFissionFragmentsCorrected": "y",
"ImpuritiesCorrected": "y",
"Mannhart": "no, but strongly recommended",
"CorrelatedData": ["31854002", "31854003", "30969002", "30969003"],
"EXFORnumber": ["22202003", "22202004", "22202005", "22202006", "22202007"],
"Accepted": "y",
"MissingCorrections": "multiple scattering",
"FissionDetector": {
    "Type": "FISCH",
    "ParticleDetected": "FF",
    "Efficiency": 99.5,
    "EfficiencyMethod": ["Calculated", "STAPRE"],
    "FFAbsorptionAngularDistributionMethod": ["Calculated", "STAPRE"],
    "AngularAcceptanceOfFF": "> 5 degree",
    "Gas": {
        "Type": "Methane",
        "Purity": 100.0,
        "Pressure": "unknown"
    },
    "Size": {
        "geometry": "unknown"
    },
    "NoPhotomultipliers": "unknown",
    "Weight": "15 g",
    "StructuralMaterial": "unknown",
    "ReferenceDetector": "N/A"
},
"NeutronDetector": {
    "Type": "SCIN (liquid)",
    "Number": "NE213",
    "ParticleDetected": "N",
    "Efficiency": 20.0,
    "ResponseDeterminationMethod": ["Calculated by MC", "NEFF4"],
    "EfficiencyDeterminationMethod": ["Calculated by MC", "NEFF4"],
```

```
            "Angle": 0.0,
            "TOFLength": {
                "Value": 2.796,
                "Unit": "m",
                "Unc": 2.0,
                "Unit_Unc": "mm"
            },
            "trsl": {
                "Value": 1.0,
                "Unit": "ns",
                "Measure": "FWHM"
            },
            "t0Determination": "Gamma Peak",
            "Gas": {
                "Type": "unknown",
                "Purity": "unknown",
                "Pressure": "unknown"
            },
            "Size": {
                "geometry": "cylinder",
                "diameter": "25.4 cm",
                "height": "5.08 cm"
            },
            "NoPhotomultipliers": 1,
            "Photomultipliers": "XP2041",
            "Weight": "unknown",
            "StructuralMaterial": "unknown",
            "ReferenceDetector": "None",
            "ReferenceReaction": "(n-p) scattering"
        },
        "Shielding": {
            "Material": "unknown",
            "Size": "unknown",
            "DistanceFromNeutronDetector": "unknown",
            "Purpose": "Background suppression"
        },
        "Sample": {
            "Activity": "1.5643e+5 f/s",
            "Reused": ["31854002", "31854003", "30969002", "30969003"],
            "Form": "layer",
            "Mass": "unknown",
            "Diameter": "15 mm",
            "Thickness": "unknown",
            "Homogeneity": "unknown",
            "Contaminants": "unknown",
            "BackingMaterial": "Pt",
            "BackingDiameter": "15 mm",
            "BackingThickness": "unknown",
            "DecayDeterminationMethod": "unknown",
```

```
            "Configuration": "OneSample",
            "Fabrication": "unknown",
            "ImpurityDetermination": "unknown"
        },
        "ShadowBar": {
            "Used": "Yes",
            "Material": ["Iron", "Water"],
            "Size": "2.4 m hollow cone",
            "Material of structure holding it": "low-mass"
        },
        "OutgoingEnergyDetermination": ["TOF", "PSD"],
        "IndividualChannelWidthDetermination": "unknown",
        "AverageChannelWidthDetermination": "unknown",
        "BackgroundMethods": {
            "General": "Shielding",
            "Wraparound": "N/A",
            "alpha": "PSD",
            "Gamma": "PSD",
            "RandomCoincidence": "Calculation"
        },
        "MultipleScatteringMethods": {
            "InSample": "Shadow bar",
            "Surrounding": "Shadow bar"
        },
        "AttenuationMethods": {
            "InSample": "Shadow bar",
            "Surrounding": "Shadow bar"
        },
        "DeadtimeDeterminationMethod": "Delayed FF detector signal to be
            stop signal"
    },
    "data": {
        "values": [0.00440083511614556, "...",  3.7058980917446034e-07],
        "uncertainties": [0.04102438299353203, "...", 0.3511438451688994],
        "correlations": [1.0, 0.7786935195966691, "...", 0.5037156088653236, 1.0],
        "structure": [
            {
                "name": "energy-in",
                "type": "pointwise",
                "limits": [[0.0]],
                "unit": "MeV"
            },
            {
                "name": "energy-out",
                "type": "pointwise",
                "limits": [[9.125, "...", 23.89]],
                "unit": "MeV"
            }
        ],
```

```
            "units": {
                "value": "1/None",
                "uncertainty": "relative"
            }
        }
    }
```

7.4. Input from ML for future curated databases

As mentioned in the requirements for the use case "Mining of Data and Metadata with Machine Learning and Artificial Intelligence", the ML analysis can feed back information into a WPEC SG-50 database. As can be seen from Figure 7.1, the ML algorithm identified clearly the data of Boldeman et al. (Boldeman et al., 1986) and Bao et al. (Bao et al., 1989) as biased around approximately 200 keV outgoing neutron energy. In addition to that, the ML algorithm indicates that the bias could be tied to ^{6}Li being used as active material in the neutron detector. The bias appears right at where the ^{6}Li(n,α) cross-section has a large resonance. If the detector response function is incorrect, one would see that strongly at this resonance. Hence, the finding of ML seems physically plausible (Walton et al., 2024). This information on potential biases can be fed back into layer three of the curated WPEC SG-50 database. Moreover, these results were used to motivate the AIACHNE measurement exploring if one could potentially correct the measurement by Boldeman et al.

8. Summary

This Working Paper establishes use cases and requirements that should guide the development of the MEDUSAL database (Machine-readable Experimental Data User App & Library), which is a curated and automatically-readable database of nuclear reaction experimental datasets built on the EXFOR database. The use cases considered are nuclear data evaluation, nuclear reaction measurement, model development, machine learning data mining and nuclear data library validation. From the goals, uses and needs of each use case, the high-level requirements, hierarchy and data container requirements were developed. Some of the EXFOR access and retrieval tools currently under development at the IAEA were described, and the additional functionality needed by one example user (the AIACHNE collaboration) was explained in detail. That additional functionality will be available in the MEDUSAL database, if it is developed and maintained. An initial draft of the specifications for the MEDUSAL database has been produced by this subgroup (SG-50, 2024) as well.

Acknowledgements

Work at Los Alamos National Laboratory (LANL) was carried out under the auspices of the National Nuclear Security Administration (NNSA) of the U.S. Department of Energy (DOE) under contract 89233218CNA000001. This material is based upon work supported by the Department of Energy National Nuclear Security Administration through the Nuclear Science and Security Consortium under Award Number(s) DE-NA0003180 and the Office of Nuclear Physics under DE-20SSC000056 and DE-SC0021243. Work at Brookhaven National Laboratory was sponsored by the Office of Nuclear Physics, Office of Science of the US Department of Energy under Contract No. DE-SC0012704 with Brookhaven Science Associates, LLC.

9. References

Bao, S., Tang, G., Fan, J., & Cao, W. (1989). Test for the corrections in the measurement of ^{252}Cf spontaneous fission prompt neutron spectrum in low energy portion. 50 Years Nuclear Fission 1989, 2, 951.

Boldeman, J. W., Clancy, B. E., & Culley, D. (1986). Measurements of the Prompt Fission Neutron Spectrum from the Spontaneous Fission of ^{252}Cf. Nuclear Science and Engineering, 93(2), 181–192. https://doi.org/10.13182/NSE86-2

Capote, R., Herman, M., Obložinský, P., et al. (2009). RIPL – Reference Input Parameter Library for Calculation of Nuclear Reactions and Nuclear Data Evaluations. Nuclear Data Sheets, 110(12), 3107–3214. https://doi.org/10.1016/j.nds.2009.10.004

Chalupka, A., Malek, L., Tagesen, S., & Böttger, R. (1990). Results of a Low-Background Measurement of the Fission Neutron Spectrum from 252Cf in the 9- to 29-MeV Energy Range. Nuclear Science and Engineering, 106(3), 367–376. https://doi.org/10.13182/NSE90-A29064

Chollet, F., Allaire, J., et al. (2017). R Interface to Keras. https://doi.org/10.32614/CRAN.package.keras

Koning, A., & Rochman, D. (2012). Modern Nuclear Data Evaluation with the TALYS Code System. Nuclear Data Sheets, 113(12), 2841–2934. https://doi.org/10.1016/j.nds.2012.11.002

Lajtai, A., Dyachenko, P., Seregina, E., & Kononov, V. (1989). New Evaluation of our Absolute Measurements of ^{252}Cf Prompt Fission Neutron Spectrum in the Low Energy Range. Physics of Neutron Emission in Fission, Proceedings of an IAEA Consultants Meeting, Mito City, Japan, 24–227 May 1988, (INDC(NDS)-220), 175–179.

Lewis, A., Neudecker, D., Carlson, A., et al. (2023). Templates of Expected Measurement Uncertainties for Capture and Charged-Particle Production Cross Section Observables. Europ. Phys. J. N, 9, 33.

Lewis, Amanda M., Carlson, Allan D., Smith, Donald L., et al. (2023). Templates of expected measurement uncertainties for total neutron cross-section observables. EPJ Nuclear Sci. Technol., 9, 34. https://doi.org/10.1051/epjn/2023018

Lovell, A. E., Mohan, A. T., Sprouse, T. M., & Mumpower, M. R. (2022). Nuclear masses learned from a probabilistic neural network. Physical Review C, 106(1), 014305. https://doi.org/10.1103/PhysRevC.106.014305

Mannhart, W. (1987). Evaluation of ^{252}Cf fission neutron spectrum between 0 MeV and 20 MeV. Proceedings of an advisory group meeting on Properties of Neutron Sources, Lenningrad, USSR, 9–13 June, 1986, (IAEA-TECHDOC-410), 158–171.

Neudecker, D., Carlson, A., Croft, S., et al. (2023). Templates of Expected Measurement Uncer--tainties for Average Prompt and Total Fission Neutron Multiplicities. Europ. Phys. J. N, 9, 30.

Neudecker, D., Devlin, M., Haight, R., et al. (2023). Templates of Expected Measurement Uncertainties for Prompt Fission Neutron Spectra. Europ. Phys. J. N, 9, 32.

Neudecker, D., Hejnal, B., Tovesson, F., et al. (2018). Template for estimating uncertainties of measured neutron-induced fission cross-sections. EPJ Nuclear Sciences & Technologies, 4(21), 21. https://doi.org/10.1051/epjn/2018026

Neudecker, D., Kelly, K., Carlson, A., Pritychenko, B., Grosskopf, M., Vander Wiel, S., & Haight, R. (2024). Re-evaluating the Prompt Fission Neutron Spectrum of Spontaneously Fissioning ^{252}Cf. In preparation.

Neudecker, D., Lewis, A., Matthews, E., et al. (2023). Templates of Expected Measurement Uncertainties. Europ. Phys. J. N, 9, 35.

Neudecker, D., Smith, D., Tovesson, F., et al. (2020). Applying a Template of Expected Uncer--tainties to Updating ^{239}Pu(n,f) Cross-section Covariances in the Neutron Data Standards Database. Nuclear Data Sheets, 163, 228–248. https://doi.org/10.1016/j.nds.2019.12.005

Neudecker, D., Taddeucci, T. N., Haight, R. C., et al. (2016). The Need for Precise and Well-documented Experimental Data on Prompt Fission Neutron Spectra from Neutron-induced Fission of ^{239}Pu. Nuclear Data Sheets, 131, 289–318. https://doi.org/10.1016/J.NDS.2015.12.005

Neudecker, D., Cabellos, O., Clark, A. R., et al. (2021). Informing nuclear physics via machine learning methods with differential and integral experiments. Physical Review C, 104(3), 1–13. https://doi.org/10.1103/PhysRevC.104.034611

Neudecker, Denise. (2018). ARIADNE - a program estimating covariances in detail for neutron experiments. EPJ Nuclear Sci. Technol., 4, 34. https://doi.org/10.1051/epjn/2018012

Okumura, S., Schnabel, G., & Koning, A. (2024). Developing a New Web Service for Experimental Nuclear Reaction Database (EXFOR) Using RESTful API and JSON. EPJ Web of Conferences, 292, 12003. https://doi.org/10.1051/epjconf/202429212003

Otuka, N., & Brown, D. (2023). Summary Report of the Technical Meeting on International Network of Nuclear Reaction Data Centres. https://doi.org/https://doi.org/10.61092/iaea.cz5x-44vw

Otuka, N., Dupont, E., Semkova, V., et al. (2014). Towards a More complete and accurate experimental nuclear reaction data library (EXFOR): International collaboration between nuclear reaction data centres (NRDC). Nuclear Data Sheets, 120, 272–276. https://doi.org/10.1016/j.nds.2014.07.065

Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

Schnabel, G. (2020). A Computational EXFOR Database. EPJ Web of Conferences, 239, 16001. https://doi.org/10.1051/epjconf/202023916001

Schwerer, O. (2015, August). EXFOR Formats Manual (tech. rep. No. IAEA-NDS-207). IAEA. Vienna.

SG-50, W. (2024). SG-50 Specifications Document (tech. rep. No. NEA No. XXXX,). NEA.

Talou, P., Stetcu, I., Jaffke, P., et al. (2021). Fission fragment decay simulations with the CGMF code. Computer Physics Communications, 269, 108087. https://doi.org/10.1016/j.cpc.2021.108087

Vanhoy, J., Haight, R., Hick, S., et al. (2023). Templates of Expected Measurement Uncertainties for (n,xn) Cross Sections. Europ. Phys. J. N, 9, 31.

Verbeke, J., Randrup, J., & Vogt, R. (2018). Fission Reaction Event Yield Algorithm FREYA 2.0.2. Computer Physics Communications, 222, 263–266. https://doi.org/https://doi.org/10.1016/j.cpc.2017.09.006

Walton, N., Neudecker, D., Wiel, S. V., Grosskopf, M., & Kelly, K. (2024). Machine-learning assisted identification of potential sources of bias in prompt-fission neutron spectra (PFNS) measurements. (To be submitted).

Whewell, B., Grosskopf, M., & Neudecker, D. (2020). Evaluating ^{239}Pu(n,f) cross sections via machine learning using experimental data, covariances, and measurement features. Nuclear Instruments and Methods in Physics Research Section A, 978, 164305. https://doi.org/10.1016/J.NIMA.2020.164305

Zerkin, V. (2024). Extended EXFOR dissemination systems for professional users, data developers and modern applications. Technical Meeting NRDC-2024 of the International Network of Nuclear Reaction Data Centres. https://nds.iaea.org/nrdc/nrdc_2024/present/zerkin1.pdf.

Zerkin, V. V. (2022). Technical Meeting NRDC-2022 of the International Network of Nuclear Reaction Data Centres. www-nds.iaea.org/nrdc/nrdc_2022

Zerkin, V. V. (2023). Progress in EXFOR/ENDF/IBANDL/PDF databases, retrieval systems, tools and software. Technical Meeting NRDC-2023 of the International Network of Nuclear Reaction Data Centres. https://www-nds.iaea.org/nrdc/nrdc_2023/present/zerkin1.pdf.

Zerkin, V. V., & Pritychenko, B. (2018). The experimental nuclear reaction data (EXFOR): Extended computer database and Web retrieval system. Nuclear Instruments and Methods in Physics Research Section A, 888, 31–43. https://doi.org/10.1016/J.NIMA.2018.01.045

Zerkin, V. (2023). X4Pro - universal, fully relational EXFOR database. EPJ Web of Conferences, 284, 14015. https://doi.org/10.1051/epjconf/202328414015